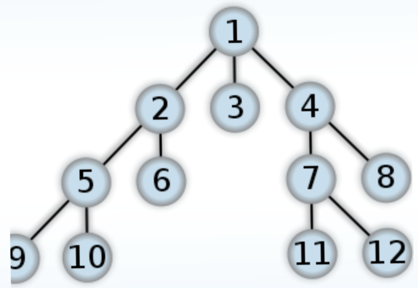



Kapitel 3: Suche in Graphen

Algorithmen und Datenstrukturen
WS 2023/24

Prof. Dr. Sándor Fekete

A & D

Algorithmus

Datenstruktur

Algorithmus 3.7

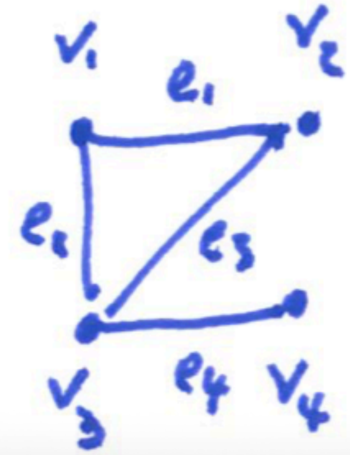
INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅

2.  WHILE (R≠∅) DO {

      2.1.  Wähle v∈R

      2.2.  IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN

            2.2.1. R:=R\{v}

      2.3.  ELSE {

            2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E

            2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

            }

    }

3.  STOP
```

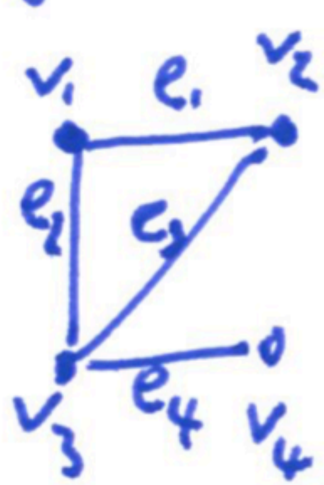
(1) Inzidenzmatrix



$$
\begin{array}{c|c}
 & \begin{matrix} e_1 & e_2 & e_3 & e_4 \end{matrix} \\
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{array}
$$

Größe: nm für einen Graphen mit n Knoten, m Kanten.

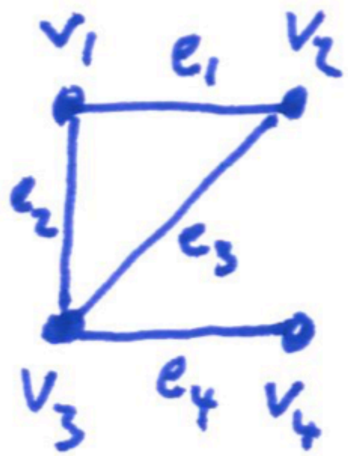
(2) Adjazenzmatrix



$$
\begin{array}{c|c}
 & \begin{matrix} v_1 & v_2 & v_3 & v_4 \end{matrix} \\
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}
\end{array}
$$

Größe: n^2 für einen Graphen mit n Knoten.

(3) Kantenliste



$$\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}$$

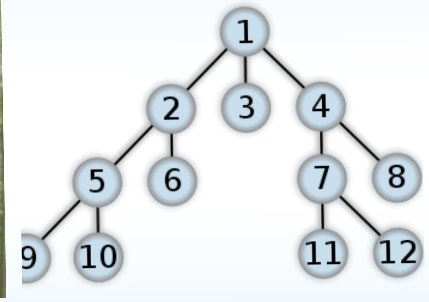
Benötigt wird eine Kantennummerierung!

$$b = \lfloor \log_2 n \rfloor + 1$$

$$d = \lfloor \log_{10} n \rfloor + 1$$

$$(2m-1) + 2m(\lfloor \log_2 n \rfloor + 1)$$

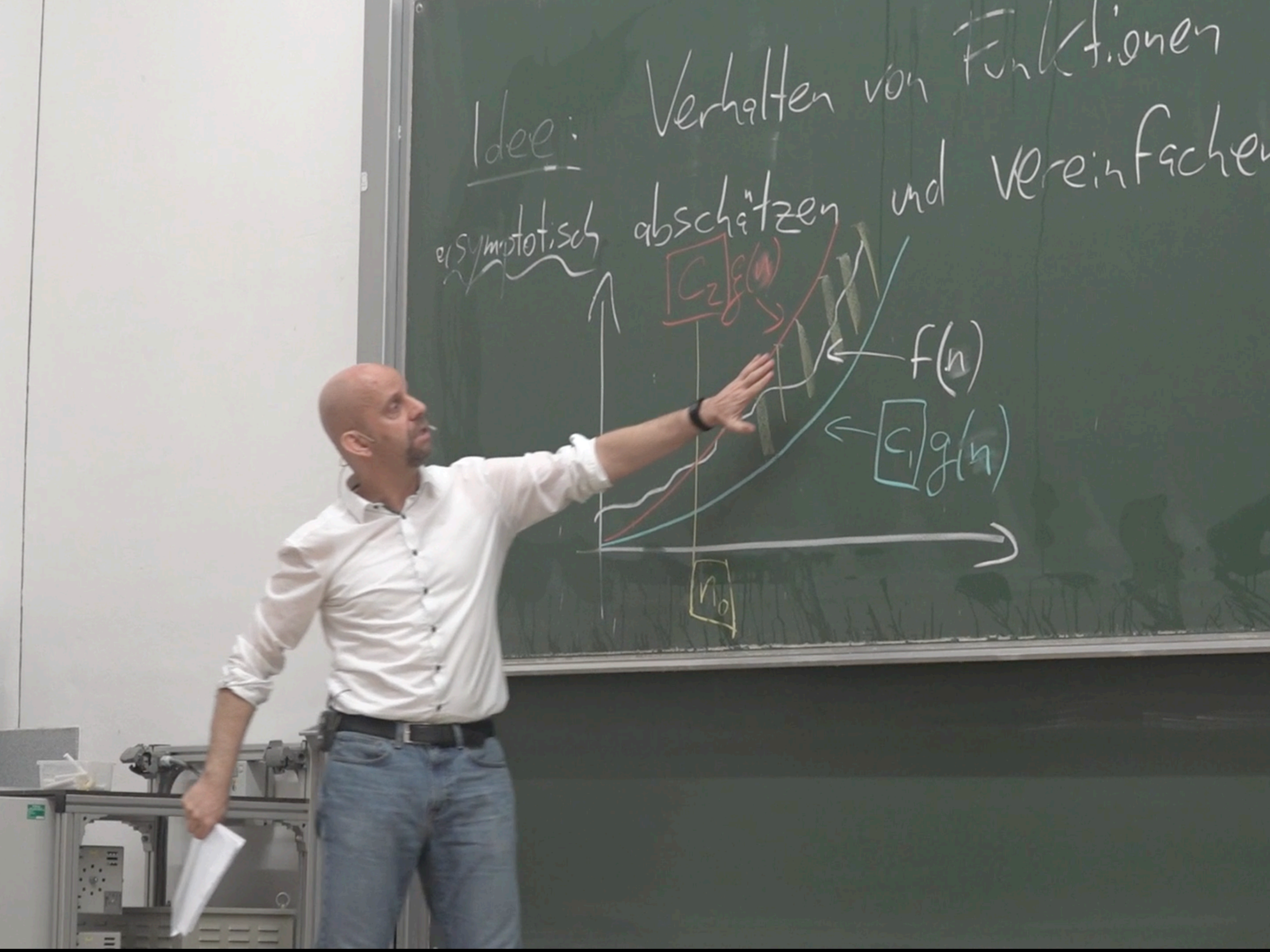
In $m \log n$ steckt das Wesen der Kantenliste

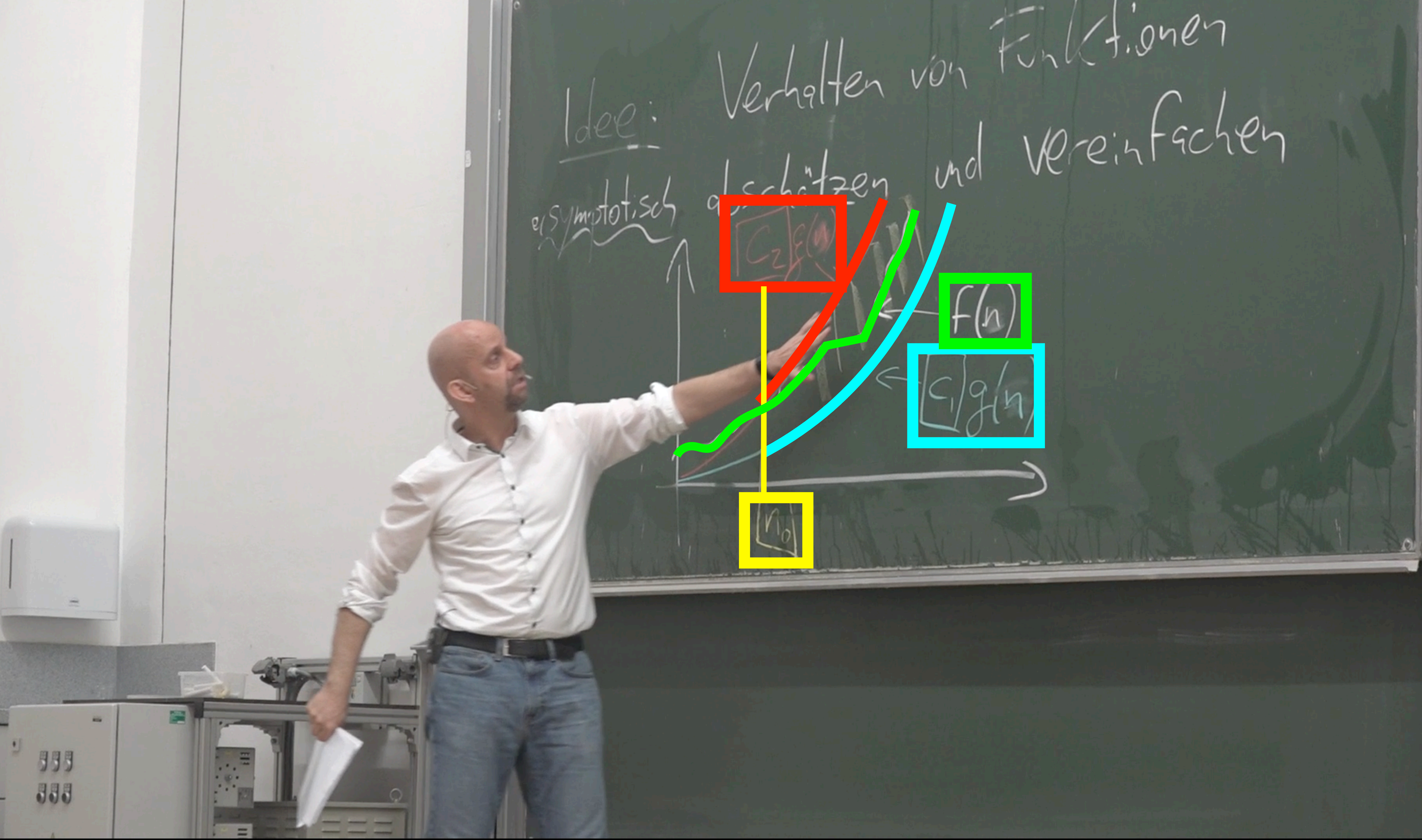


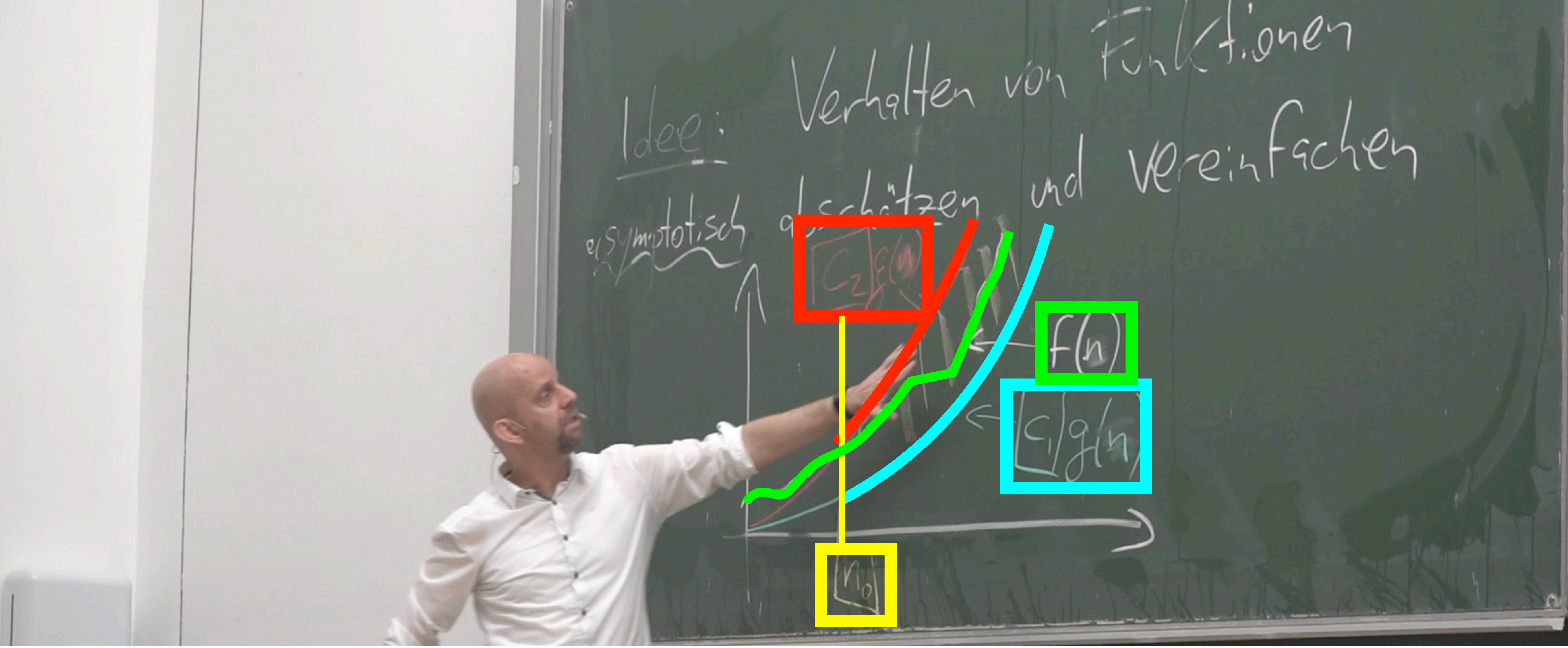
Kapitel 3.7: Wachstum von Funktionen

Algorithmen und Datenstrukturen
WS 2023/24

Prof. Dr. Sándor Fekete


Idee: Verhalten von Funktionen
asymptotisch abschätzen und vereinfachen
c2 g(n)
f(n)
c1 g(n)
n0


Verhalten von Funktionen
Idee:
asymptotisch abschätzen und vereinfachen
c2 g(n)
f(n)
c1 g(n)
n0



DEFINITION 3.9 (Θ-Notation)

Seien $f, g : \mathbb{N} \to \mathbb{R}$ Funktionen.

Dann gilt

$f \in \Theta(g) :\Leftrightarrow$ Es gibt positive Konstanten c_1, c_2, n_0 mit
$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ für alle $n \geq n_0$

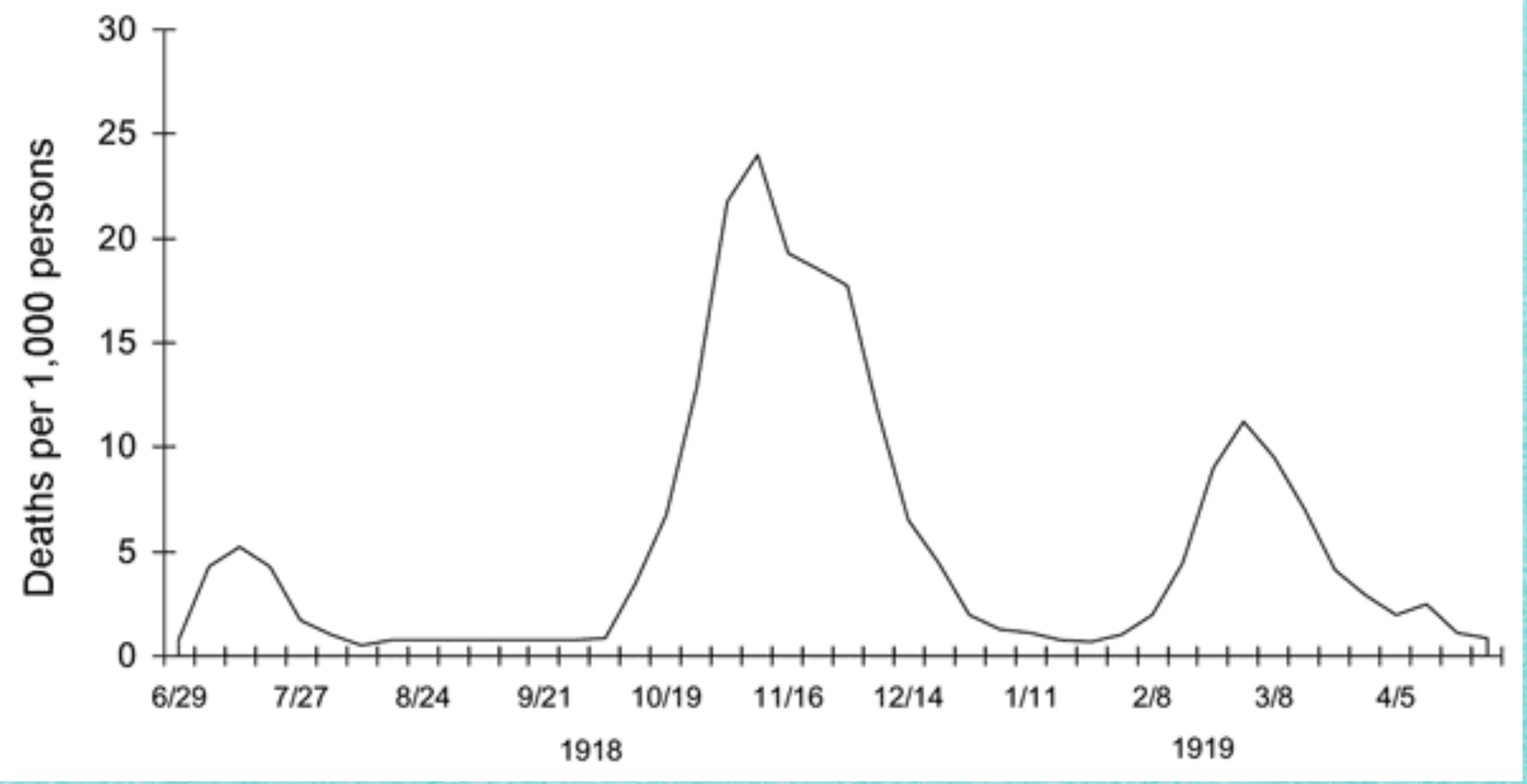
Man sagt: f wächst asymptotisch in derselben Größenordnung wie g.

Vorlesung 9

- **Datum:** Mittwoch, 27.11.2019
- **Inhalt:** Wachstum von Funktionen; O-Notation
- **Notizen:** HIER (PDF, 1.0MB)
- **Weitere Links:**
  Wikipedia-Seite: Spieltheorie
  Wikipedia-Seite: Gefangenendilemma
  Die tanzenden Roboter
  Die Theorie dahinter

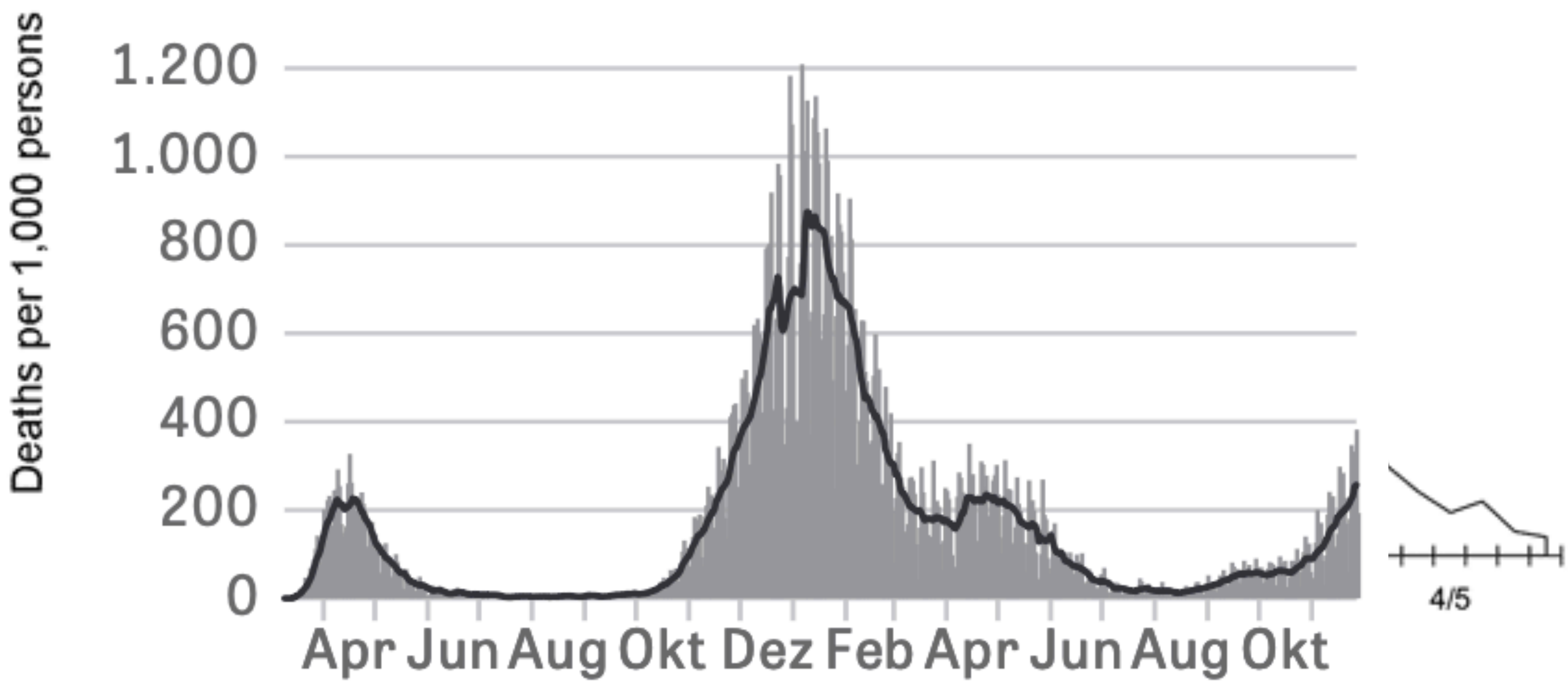
(Für Tonspur auf Bild klicken - und über den Unterschied von t^2 und 2^t nachdenken: Der Aufwand vervierfacht sich nicht nur, sondern quadriert sich!)

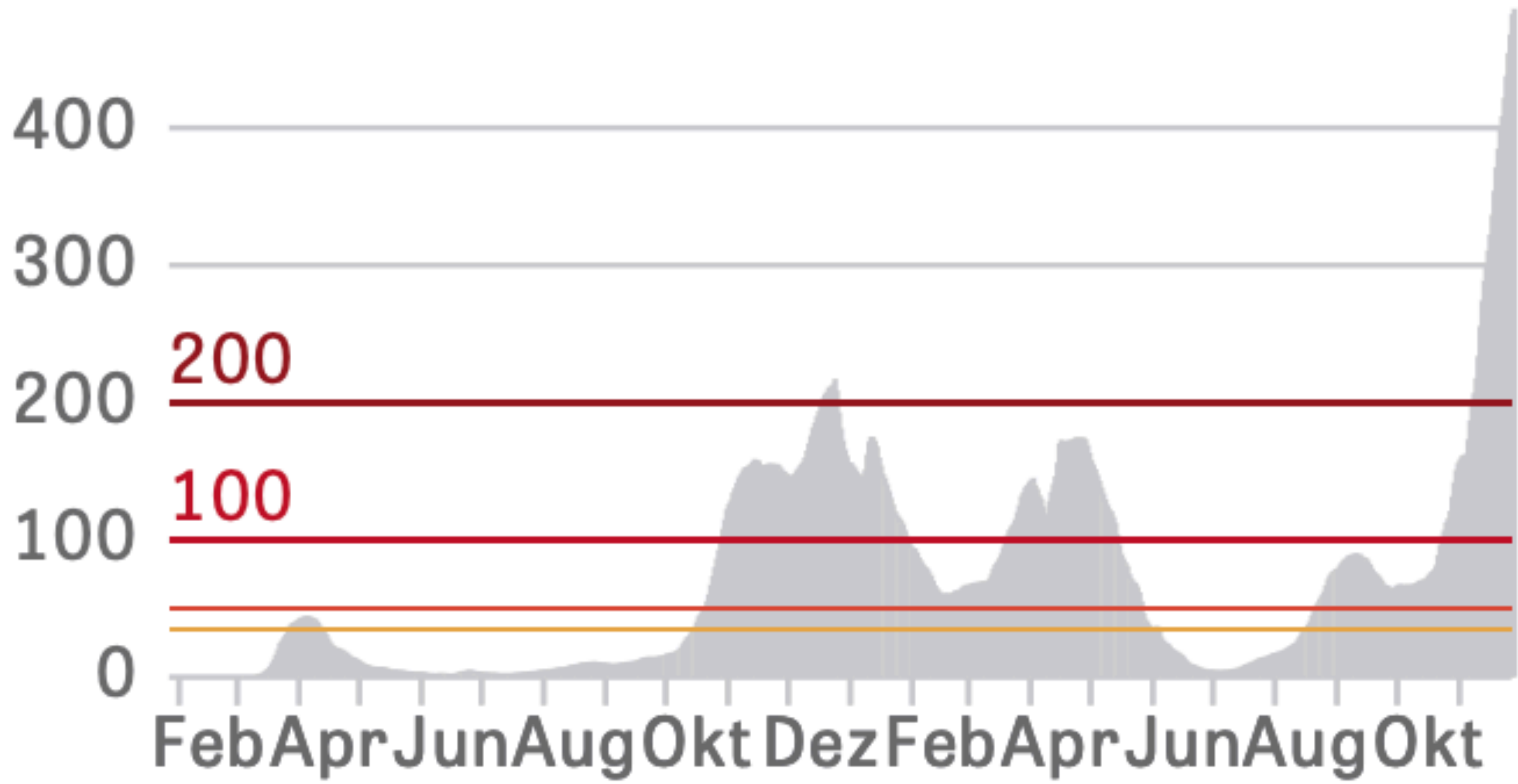
Wikipedia zum Ersten Weltkrieg
Wikipedia zur Spanischen Grippe
Wikipedia zur Pest
Wikipedia zur asymptotischen Notation

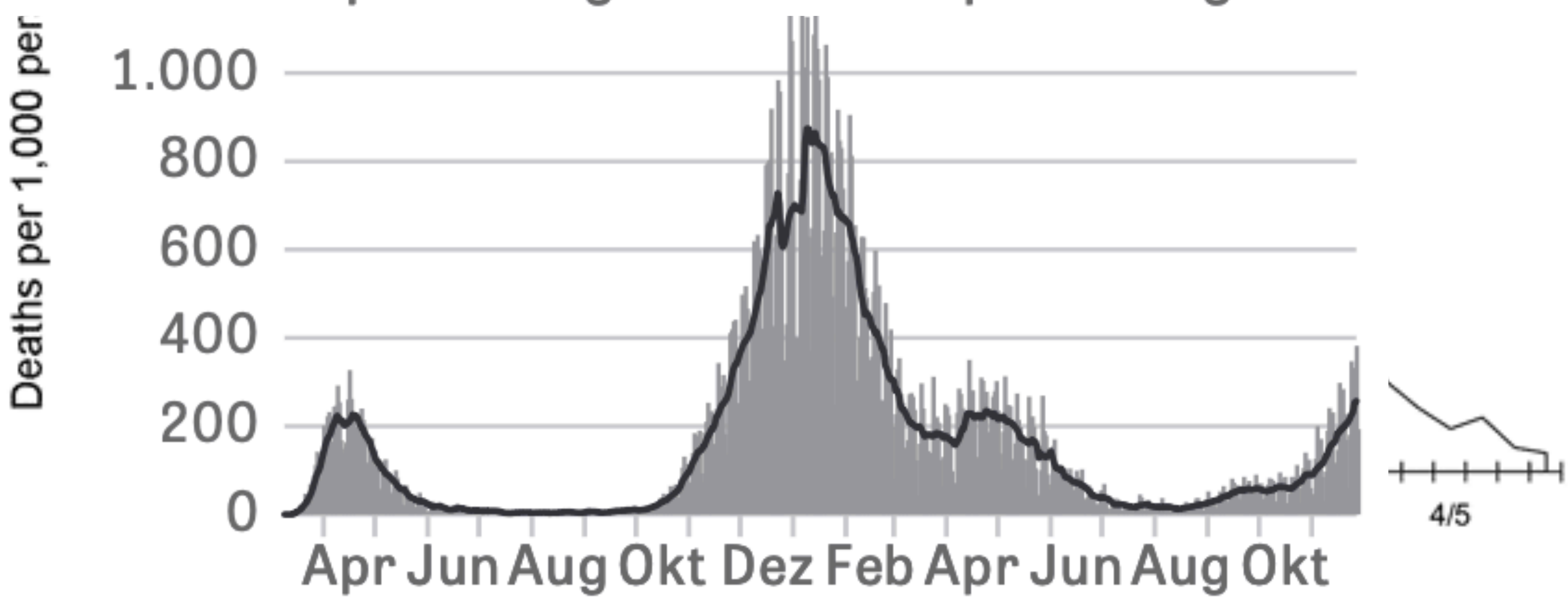


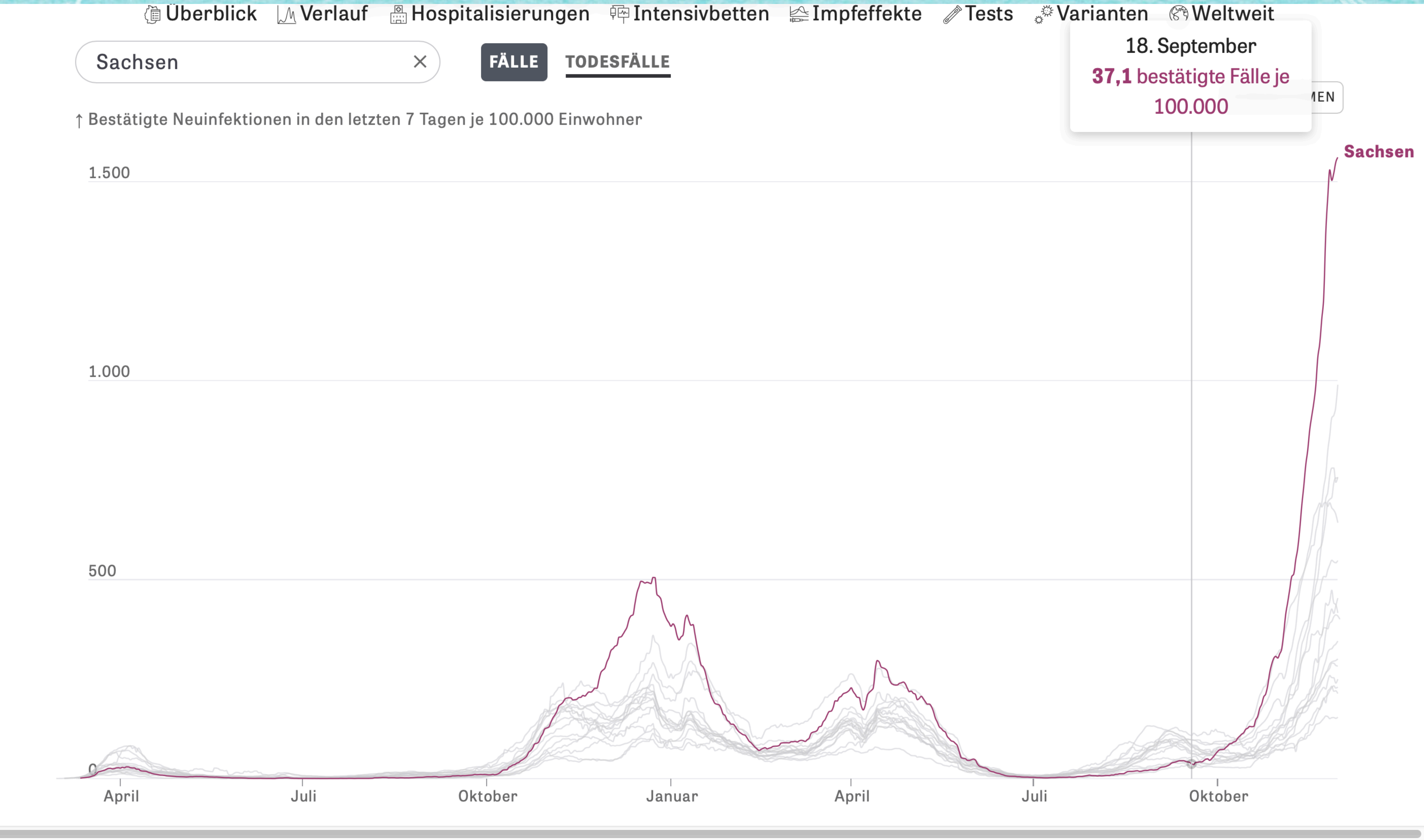
Deaths per 1,000 persons
30
25
20
15
10
5
0
6/29
7/27
8/24
9/21
10/19
11/16
12/14
1/11
2/8
3/8
4/5
1918
1919

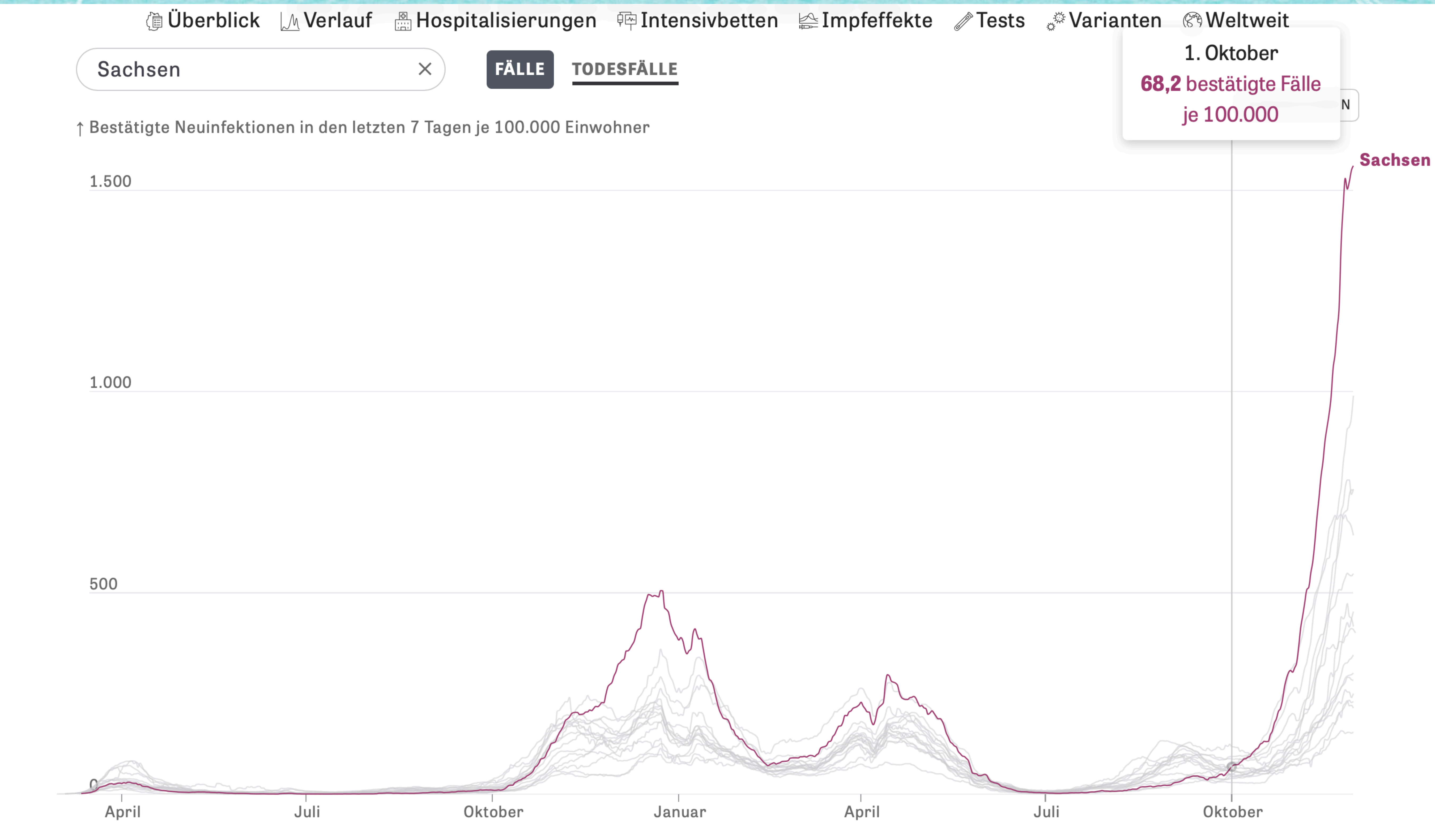
Todesfälle pro Tag




400
300
200
200
100
100
0
Feb Apr Jun Aug Okt Dez Feb Apr Jun Aug Okt


Deaths per 1,000 per
1.000
800
600
400
200
0
Apr Jun Aug Okt Dez Feb Apr Jun Aug Okt
4/5

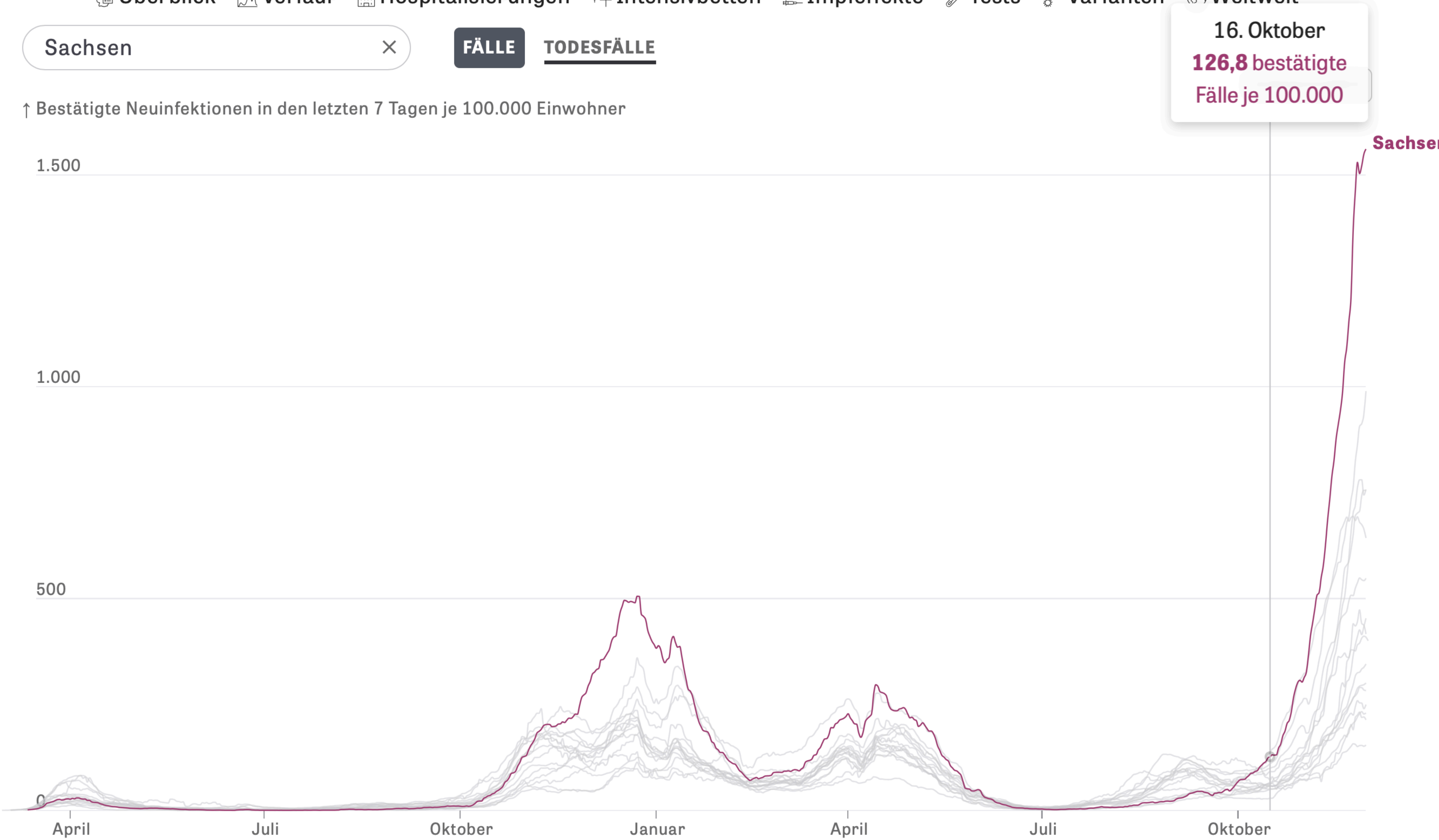

Überblick
Verlauf
Hospitalisierungen
Intensivbetten
Impfeffekte
Tests
Varianten
Weltweit
Sachsen
FÄLLE
TODESFÄLLE
18. September
37,1 bestätigte Fälle je 100.000
↑ Bestätigte Neuinfektionen in den letzten 7 Tagen je 100.000 Einwohner
Sachsen
1.500
1.000
500
0
April
Juli
Oktober
Januar
April
Juli
Oktober

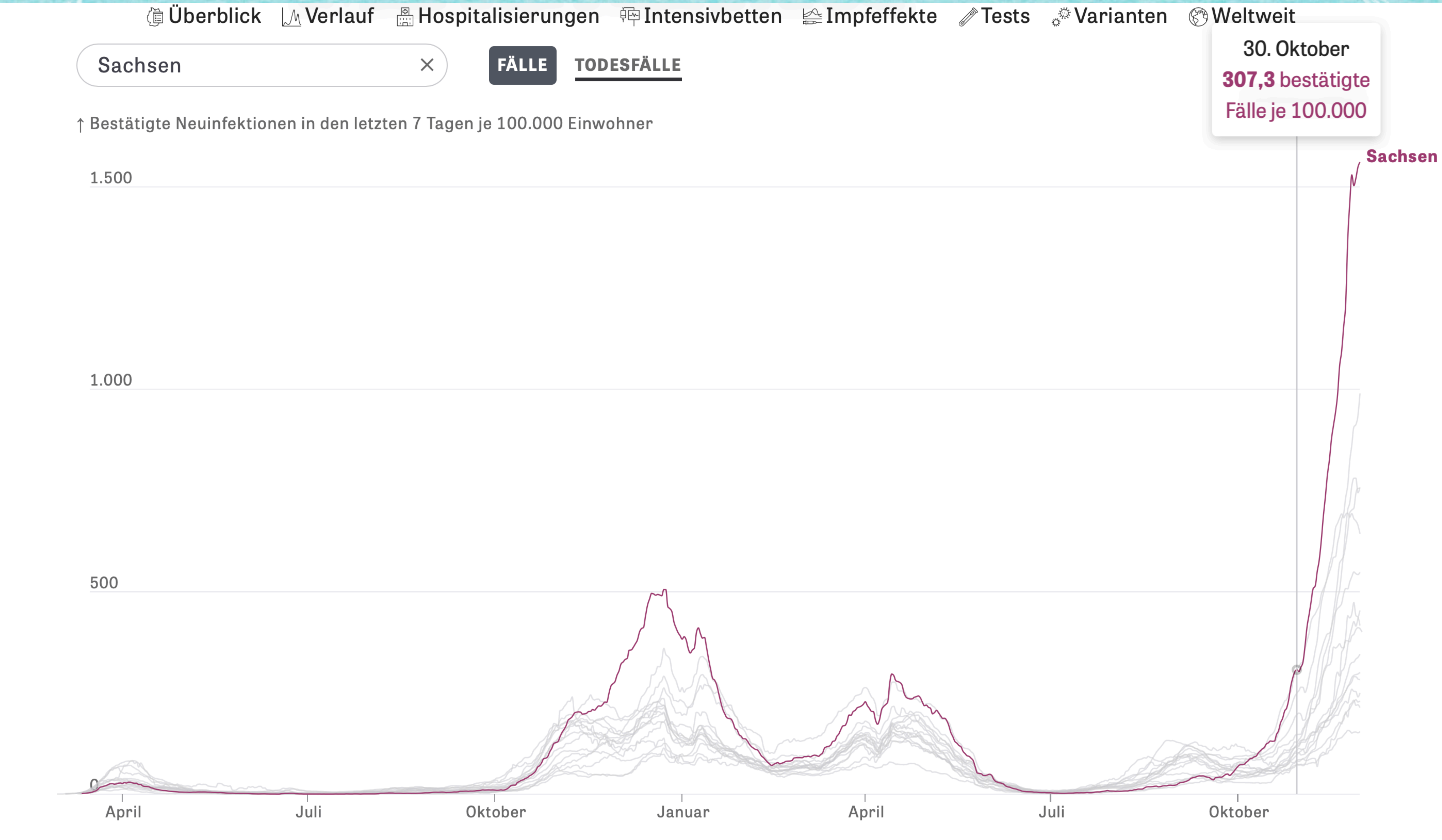

Überblick
Verlauf
Hospitalisierungen
Intensivbetten
Impfeffekte
Tests
Varianten
Weltweit
Sachsen
FÄLLE
TODESFÄLLE
1. Oktober
68,2 bestätigte Fälle
je 100.000
↑ Bestätigte Neuinfektionen in den letzten 7 Tagen je 100.000 Einwohner
Sachsen
1.500
1.000
500
0
April
Juli
Oktober
Januar
April
Juli
Oktober

Überblick Verlauf Hospitalisierungen Intensivbetten Impfeffekte Tests Varianten Weltweit

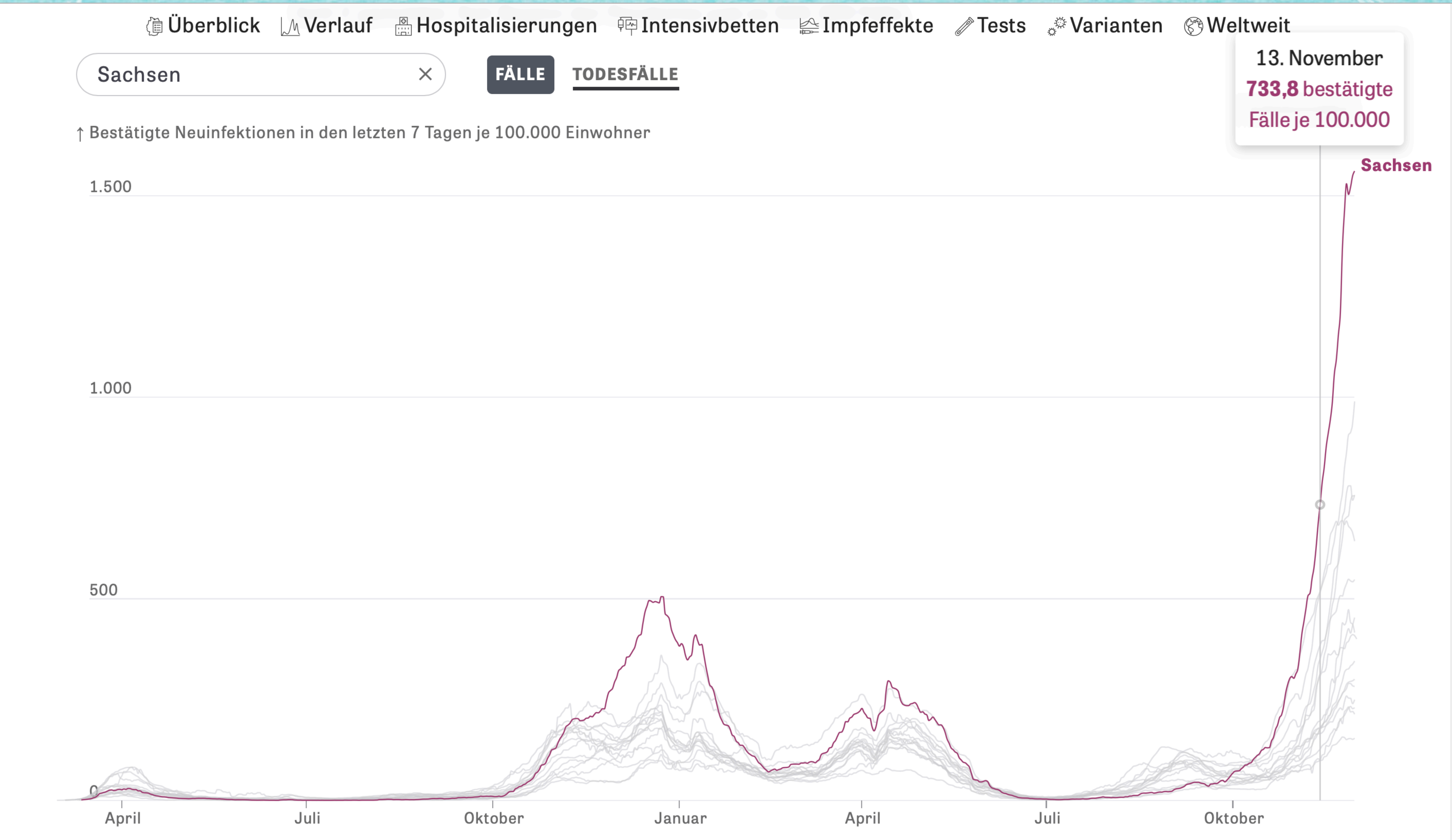
Sachsen

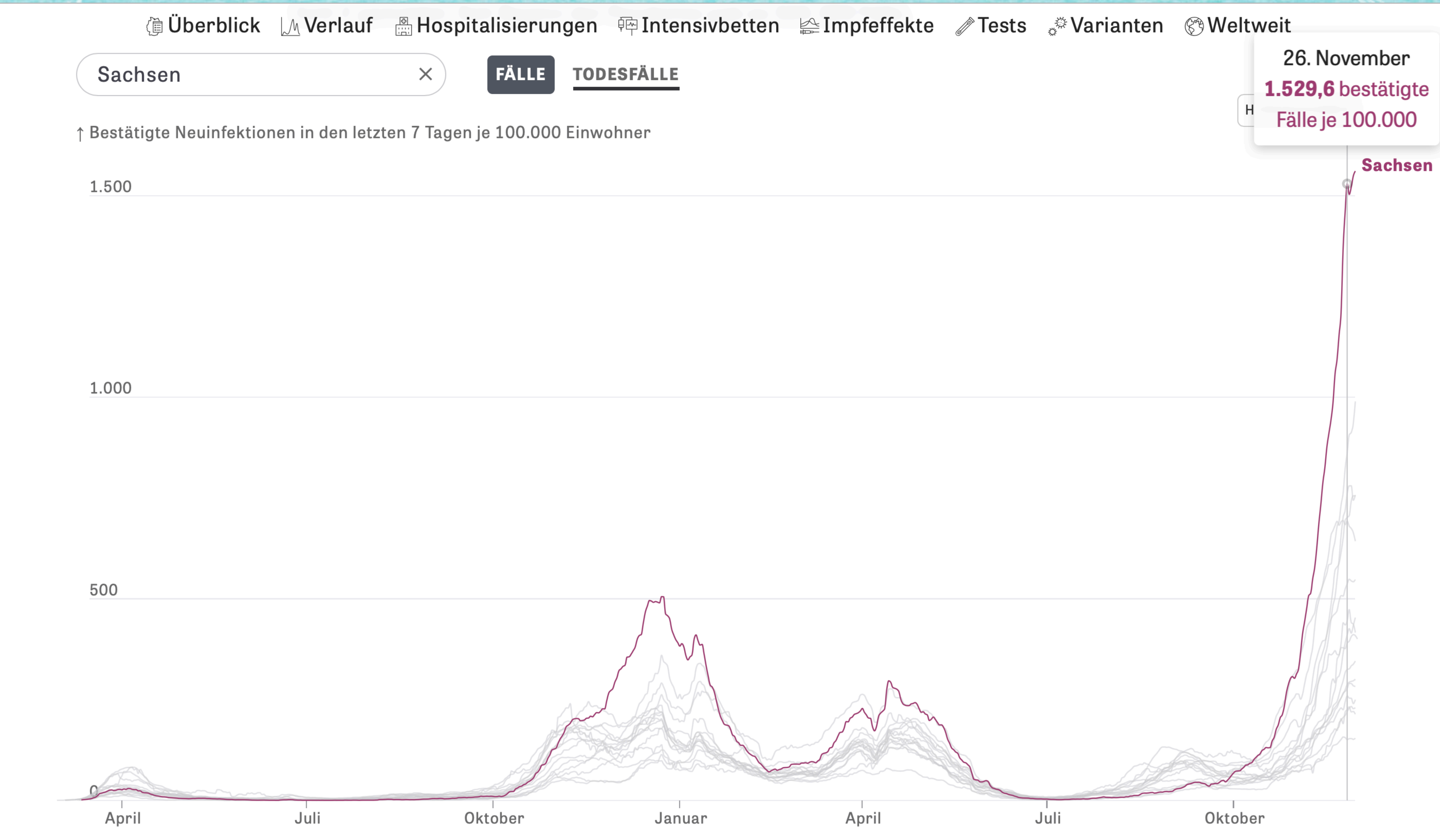
FÄLLE TODESFÄLLE



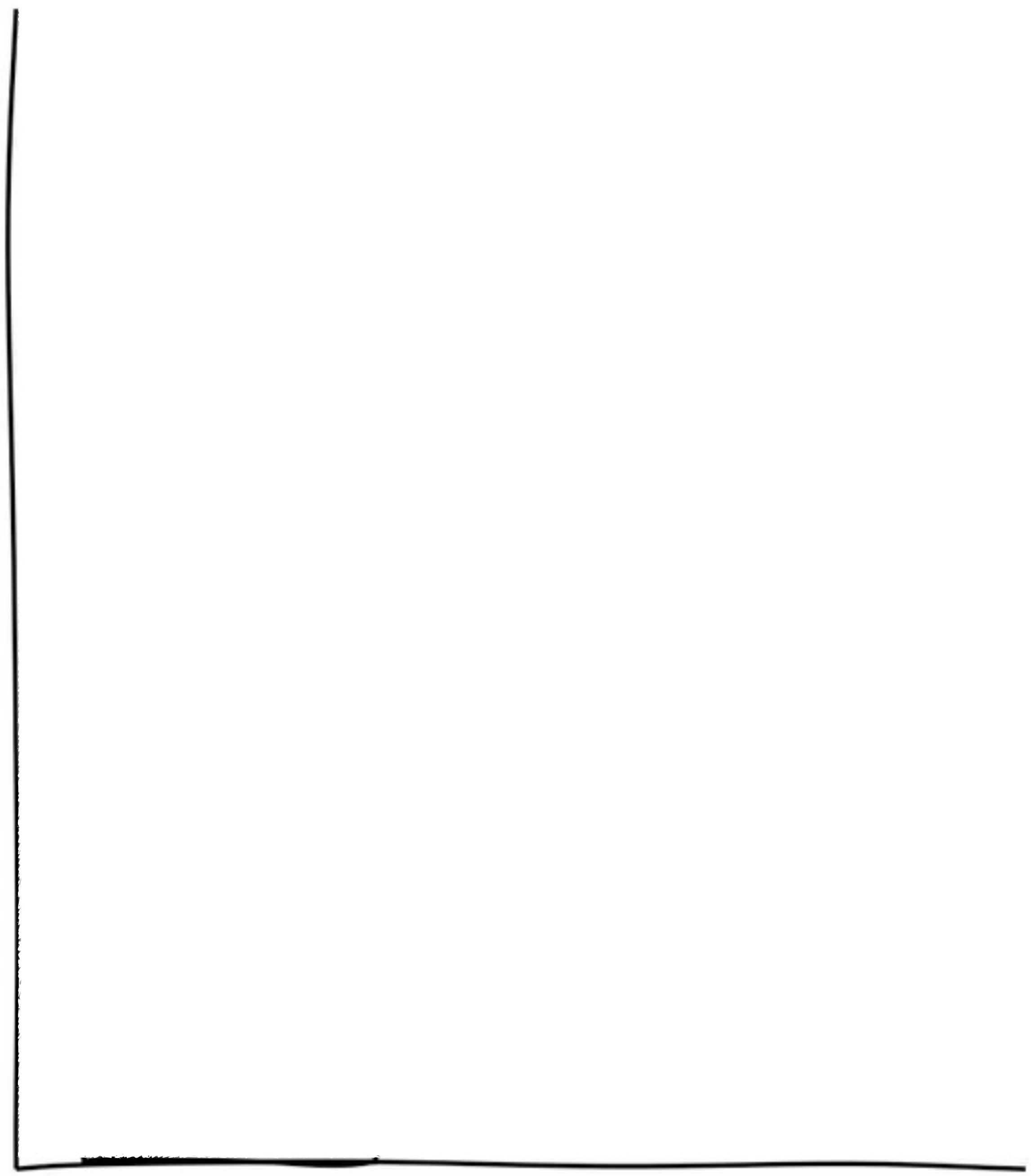


Überblick
Verlauf
Hospitalisierungen
Intensivbetten
Impfeffekte
Tests
Varianten
Weltweit
Sachsen
FÄLLE
TODESFÄLLE
30. Oktober
307,3 bestätigte
Fälle je 100.000
↑ Bestätigte Neuinfektionen in den letzten 7 Tagen je 100.000 Einwohner
Sachsen
1.500
1.000
500
0
April
Juli
Oktober
Januar
April
Juli
Oktober

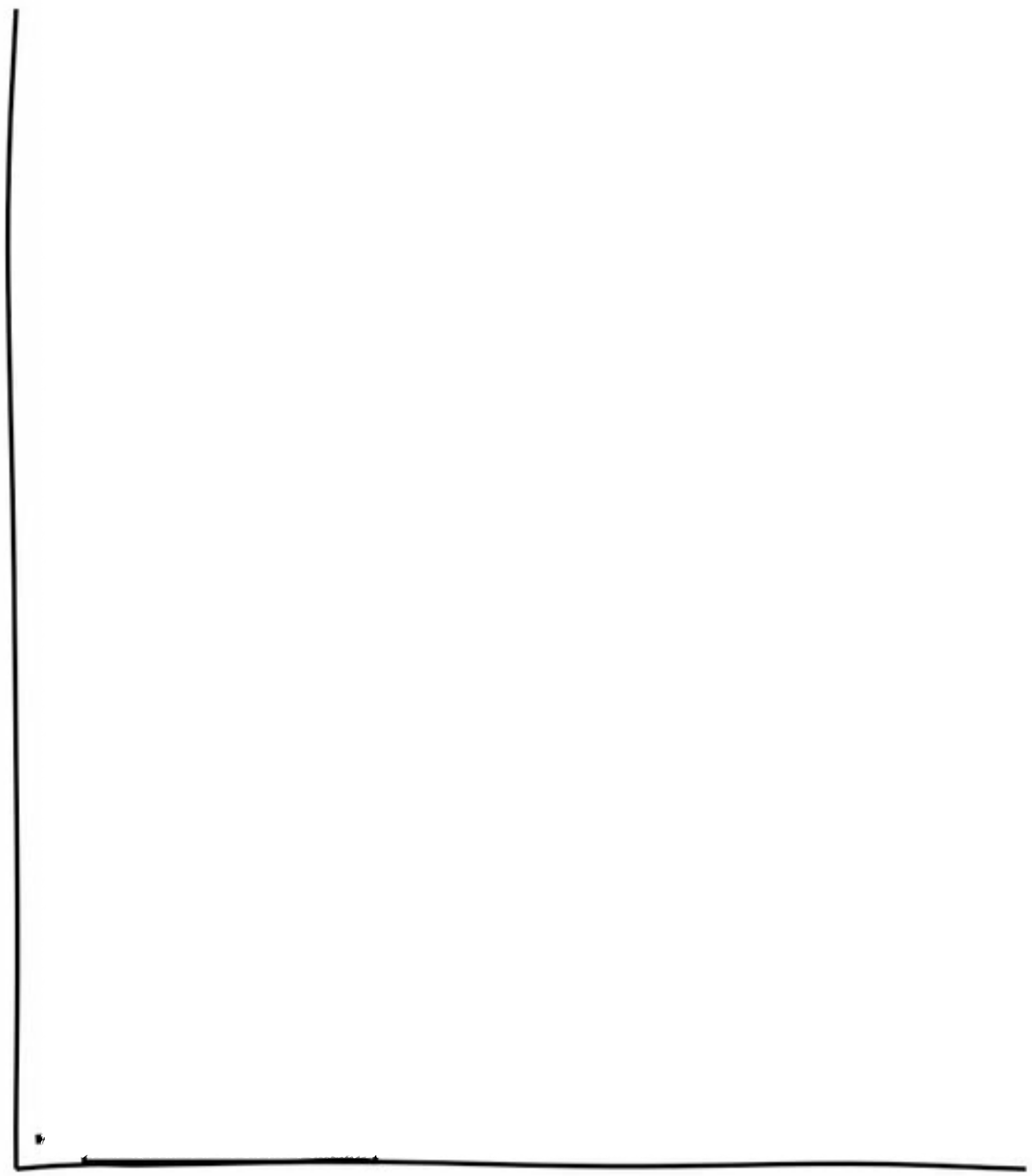

Überblick
Verlauf
Hospitalisierungen
Intensivbetten
Impfeffekte
Tests
Varianten
Weltweit
Sachsen
FÄLLE
TODESFÄLLE
13. November
733,8 bestätigte
Fälle je 100.000
↑ Bestätigte Neuinfektionen in den letzten 7 Tagen je 100.000 Einwohner
Sachsen
1.500
1.000
500
0
April
Juli
Oktober
Januar
April
Juli
Oktober

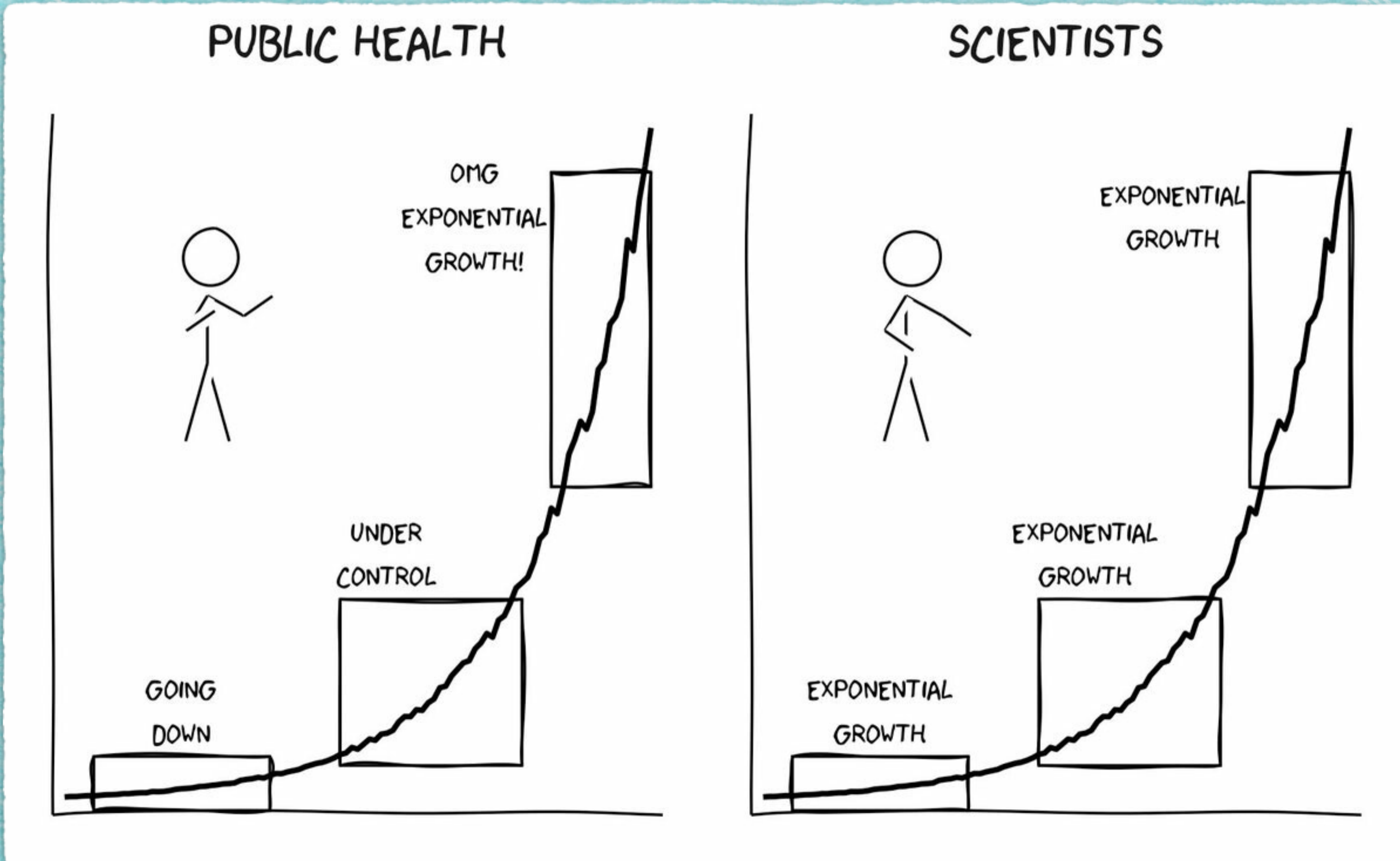

Überblick
Verlauf
Hospitalisierungen
Intensivbetten
Impfeffekte
Tests
Varianten
Weltweit
Sachsen
FÄLLE
TODESFÄLLE
26. November
1.529,6 bestätigte Fälle je 100.000
↑ Bestätigte Neuinfektionen in den letzten 7 Tagen je 100.000 Einwohner
Sachsen
1.500
1.000
500
0
April
Juli
Oktober
Januar
April
Juli
Oktober

PUBLIC HEALTH



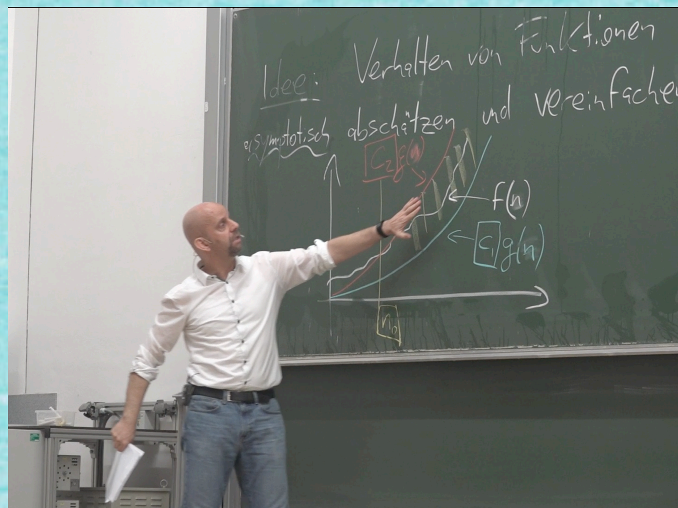
SCIENTISTS





PUBLIC HEALTH
OMG EXPONENTIAL GROWTH!
UNDER CONTROL
GOING DOWN
SCIENTISTS
EXPONENTIAL GROWTH
EXPONENTIAL GROWTH
EXPONENTIAL GROWTH

Wofür ist das gut?

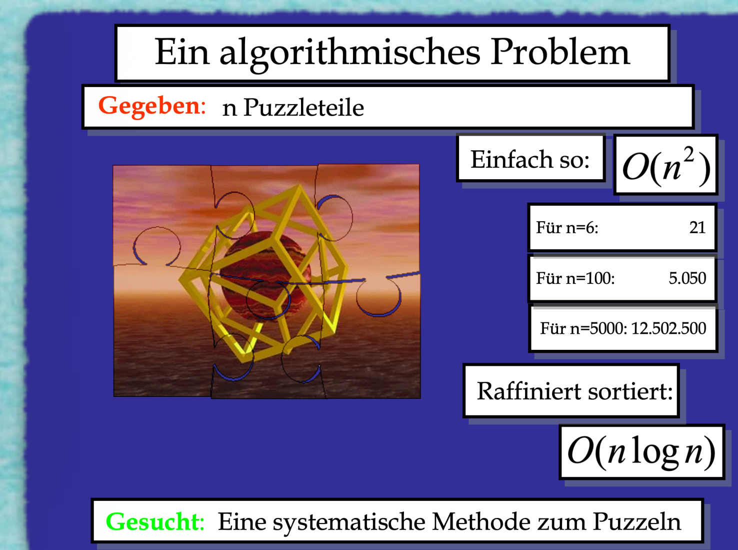


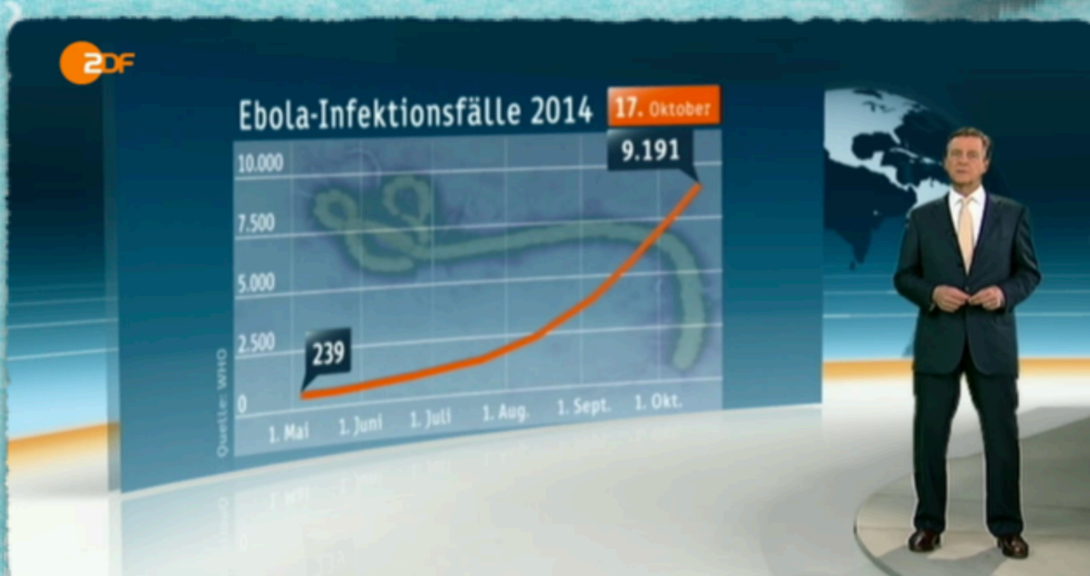
$$\log_2 \lfloor (2n + 4m + n(\lfloor \log_2 n \rfloor + 1) + 2m(\lfloor \log_2 n \rfloor + 1)) \rfloor + 1$$

$$\Theta(m \log n)$$

Einfach so: $O(n^2)$

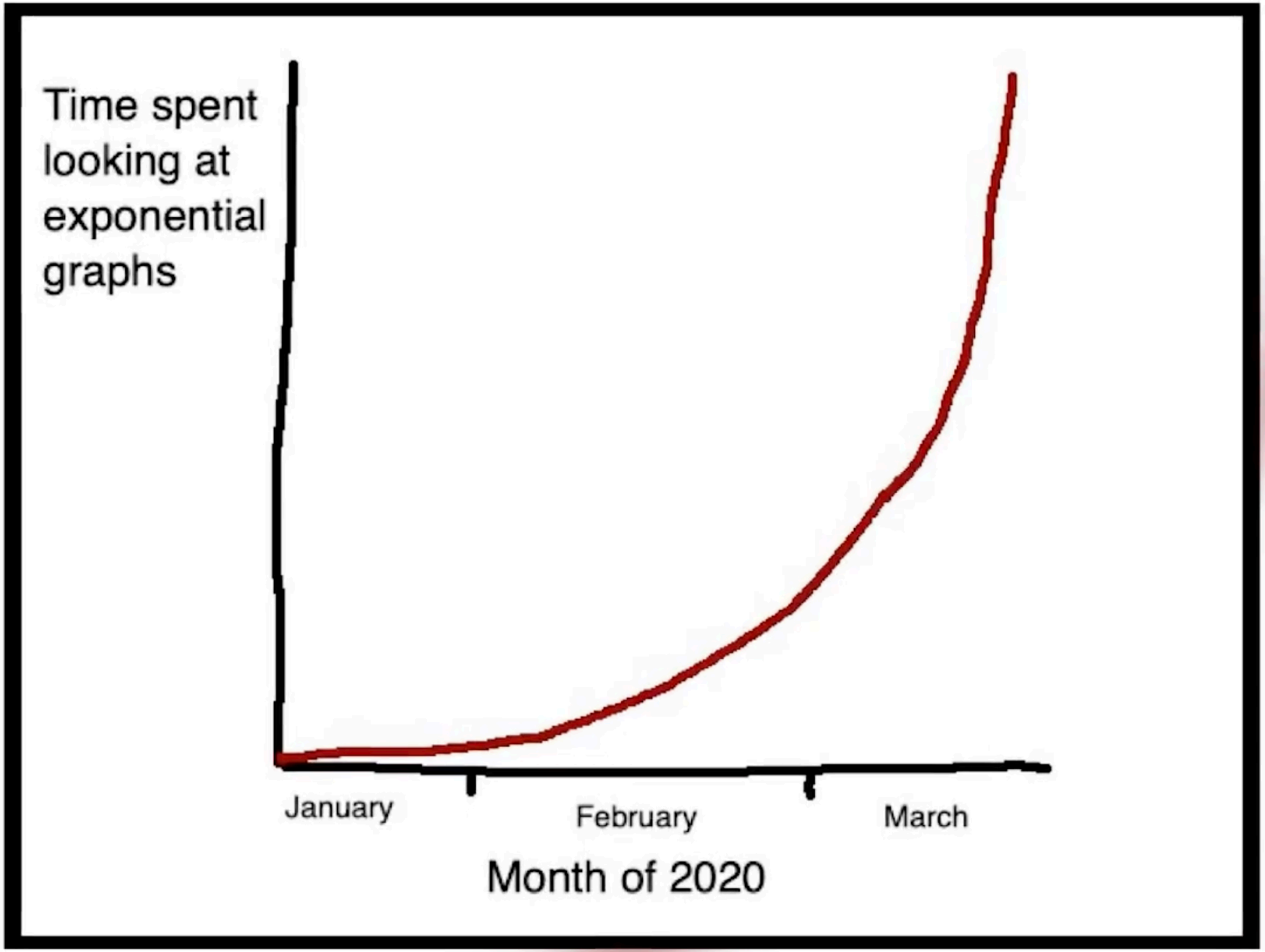
Für n=6:	21
Für n=100:	5.050
Für n=5000:	12.502.500



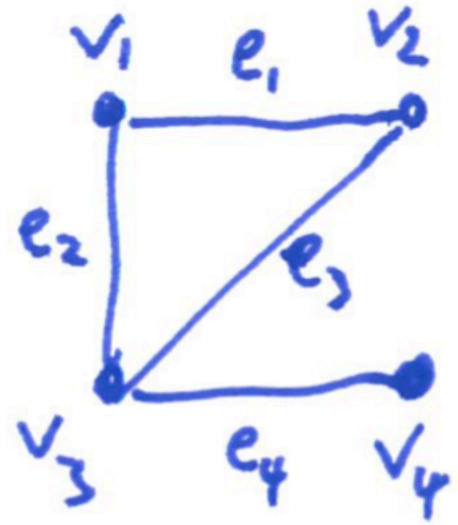


$$\Theta(n^2) \qquad (2n)^2 = 4 \cdot n^2$$

$$\Theta(2^n) \qquad 2^{2n} = 2^n \cdot 2^n$$


Time spent looking at exponential graphs
January
February
March
Month of 2020

(4) Adjazenzliste



v_1 : v_2 , v_3 ;

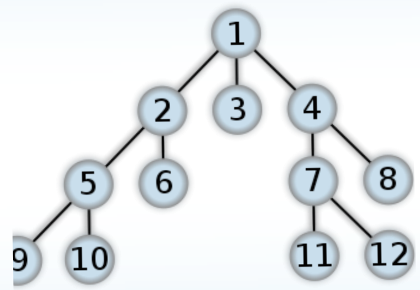
v_2 : v_1 , v_3 ;

v_3 : v_1 , v_2 , v_4 ;

v_4 : v_3 ;

Mehr an der Tafel!

s.fekete@tu-bs.de



Kapitel 3.8: Laufzeit von DFS und BFS

Algorithmen und Datenstrukturen
WS 2021/22

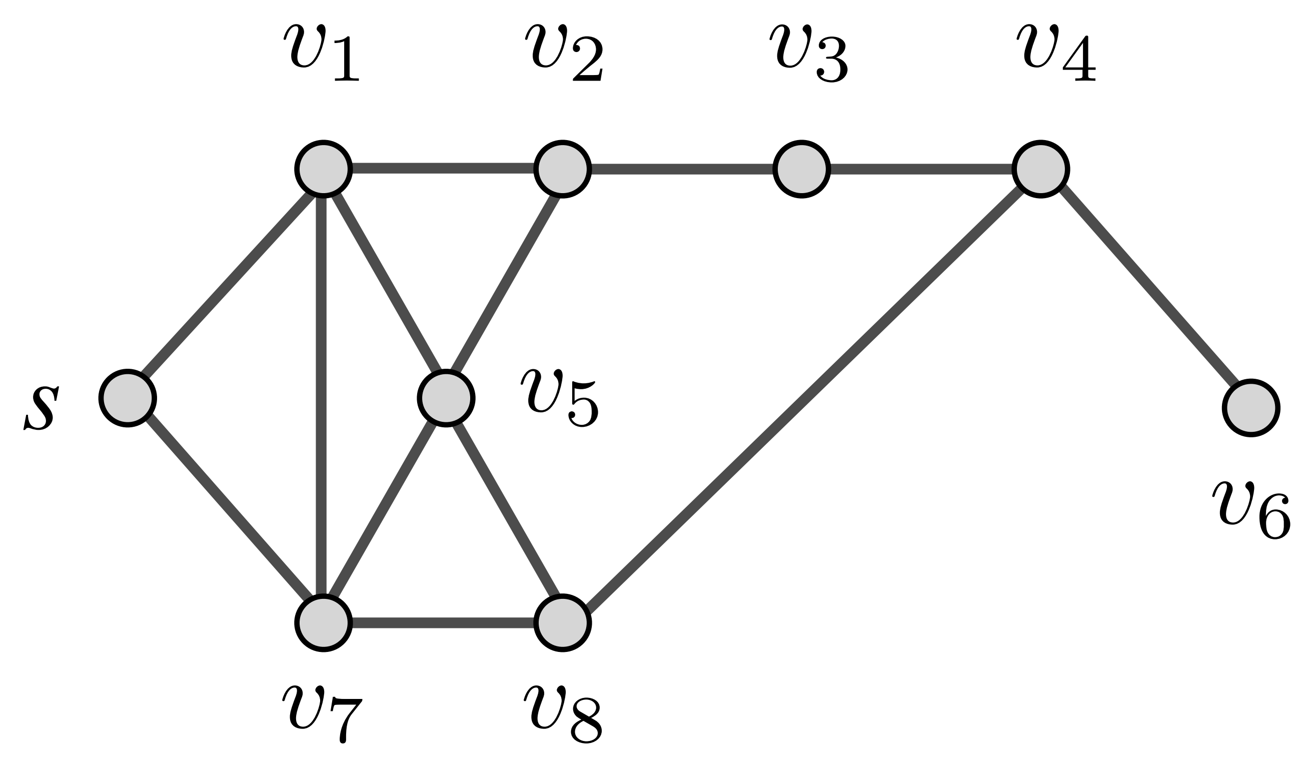
Prof. Dr. Sándor Fekete

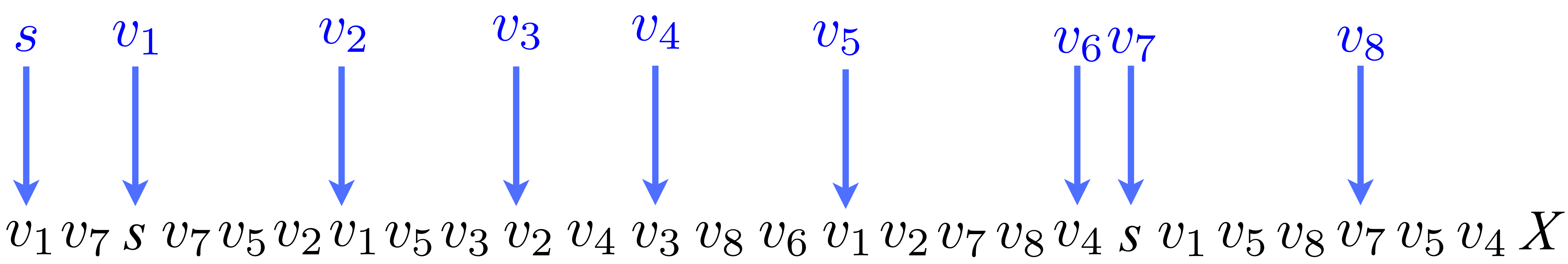
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

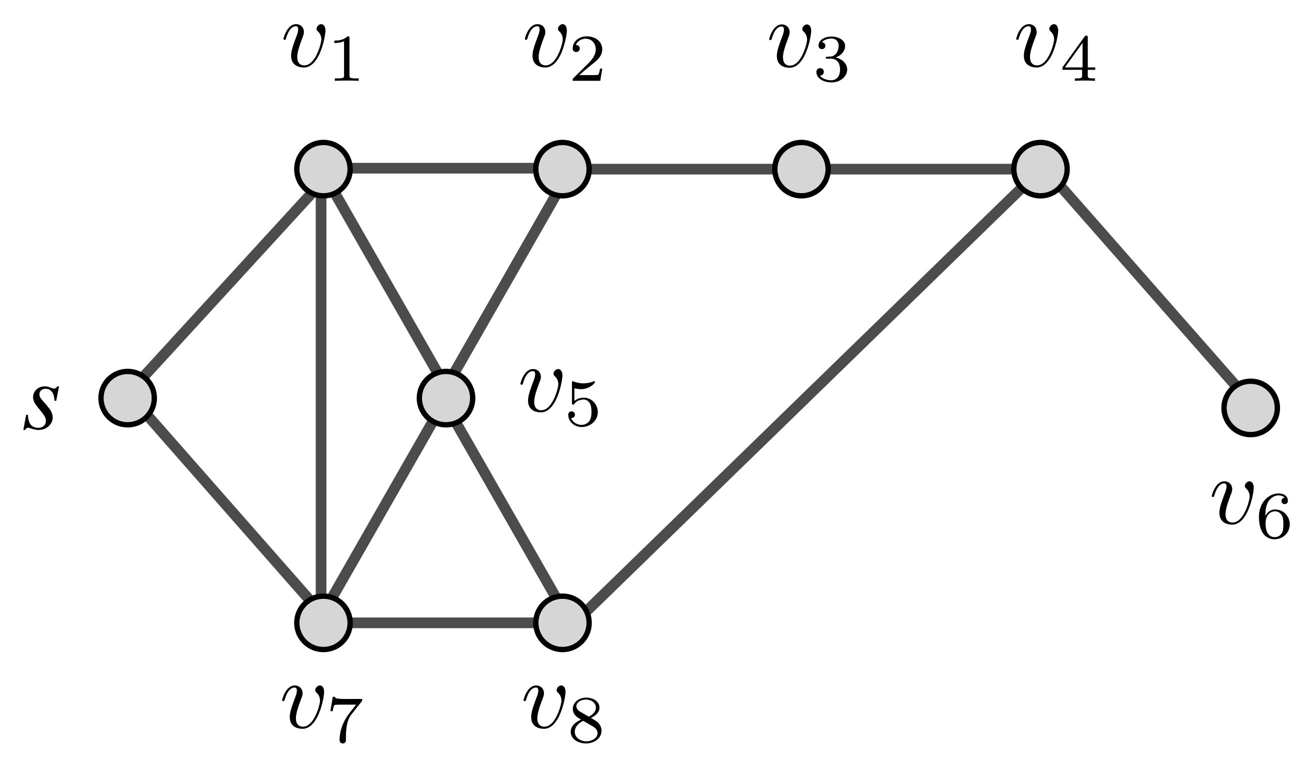
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

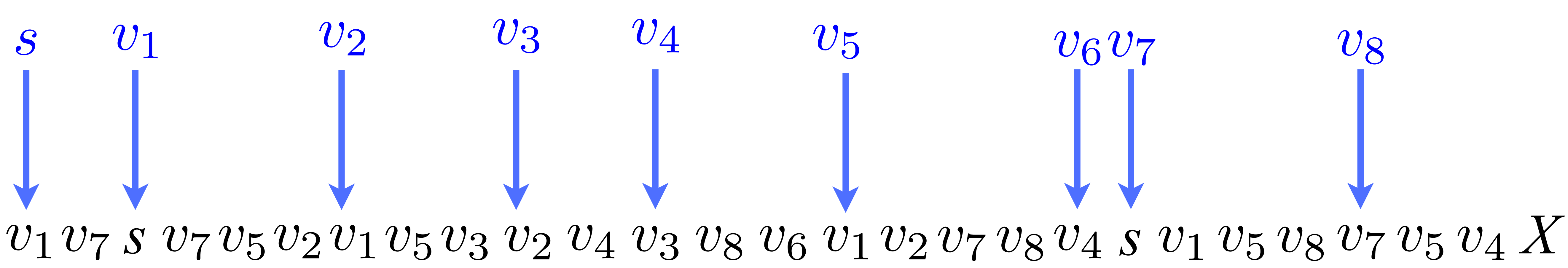
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

Sei R:={s}, Y:={s}, T:=∅

2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

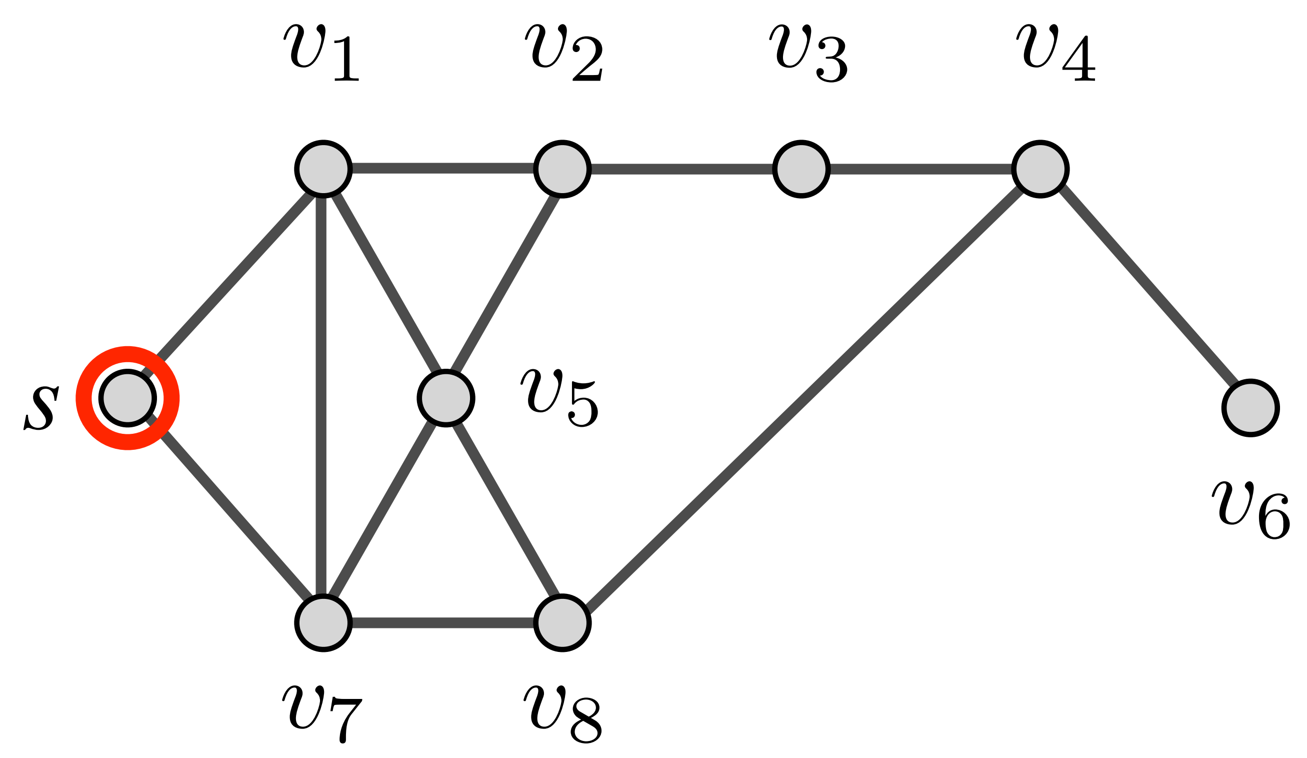
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E

        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }

3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \, v_7 \quad v_8$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5\, v_8\, v_7\, v_5\, v_4\, X$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

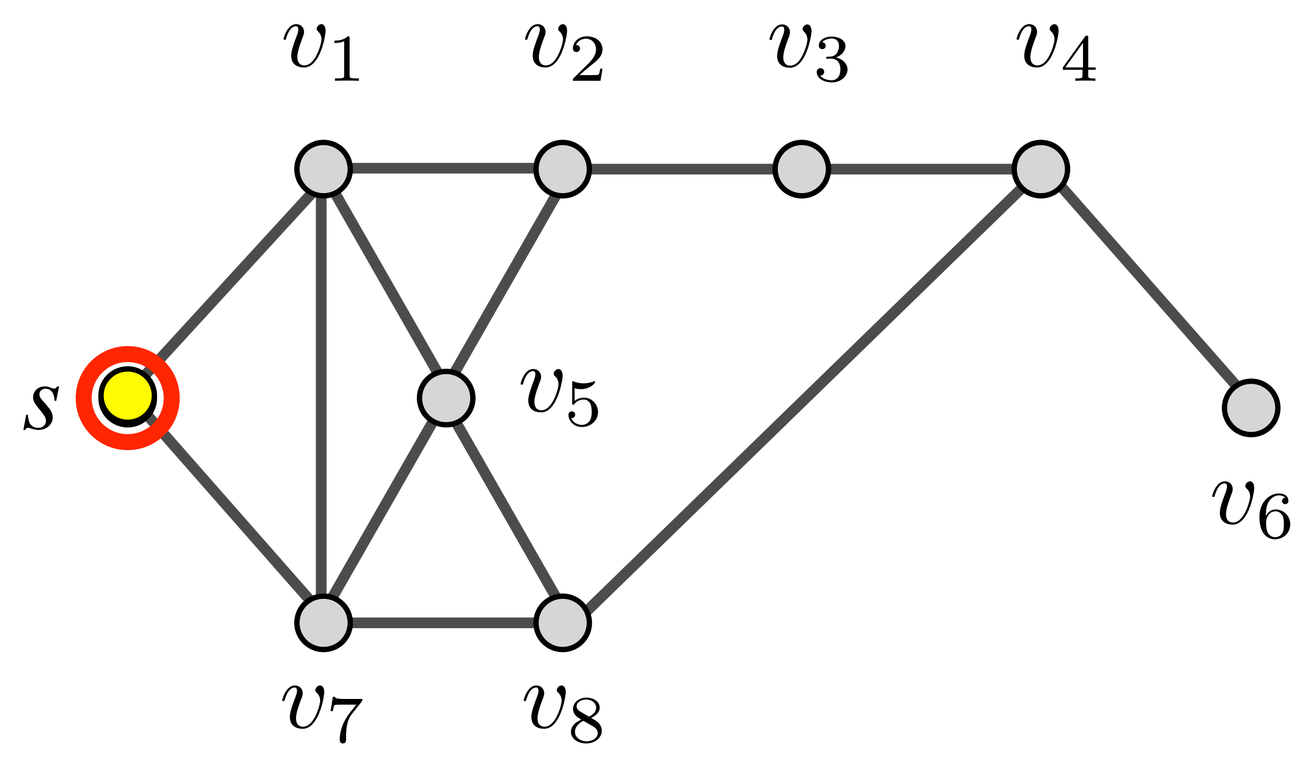
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

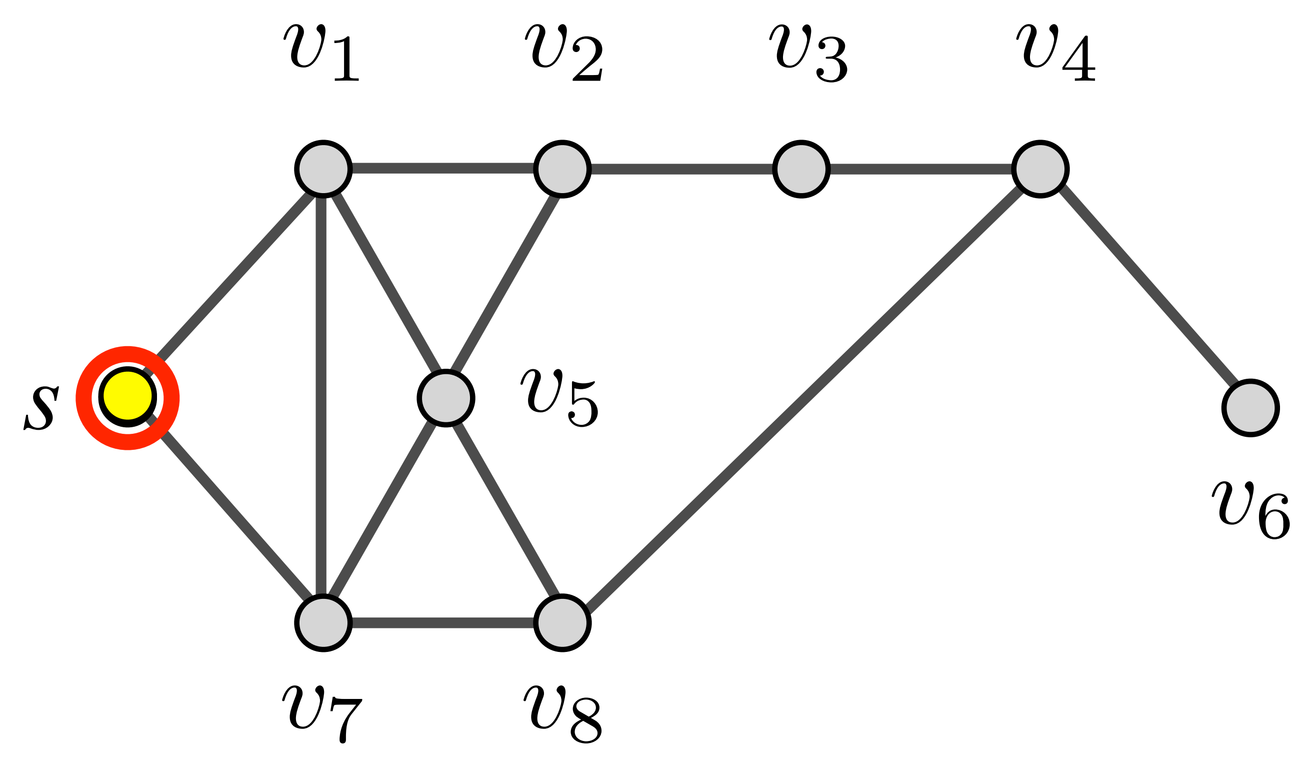
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP



$s \quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6\, v_7 \quad v_8$

$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5\, v_8\, v_7\, v_5\, v_4\, X$

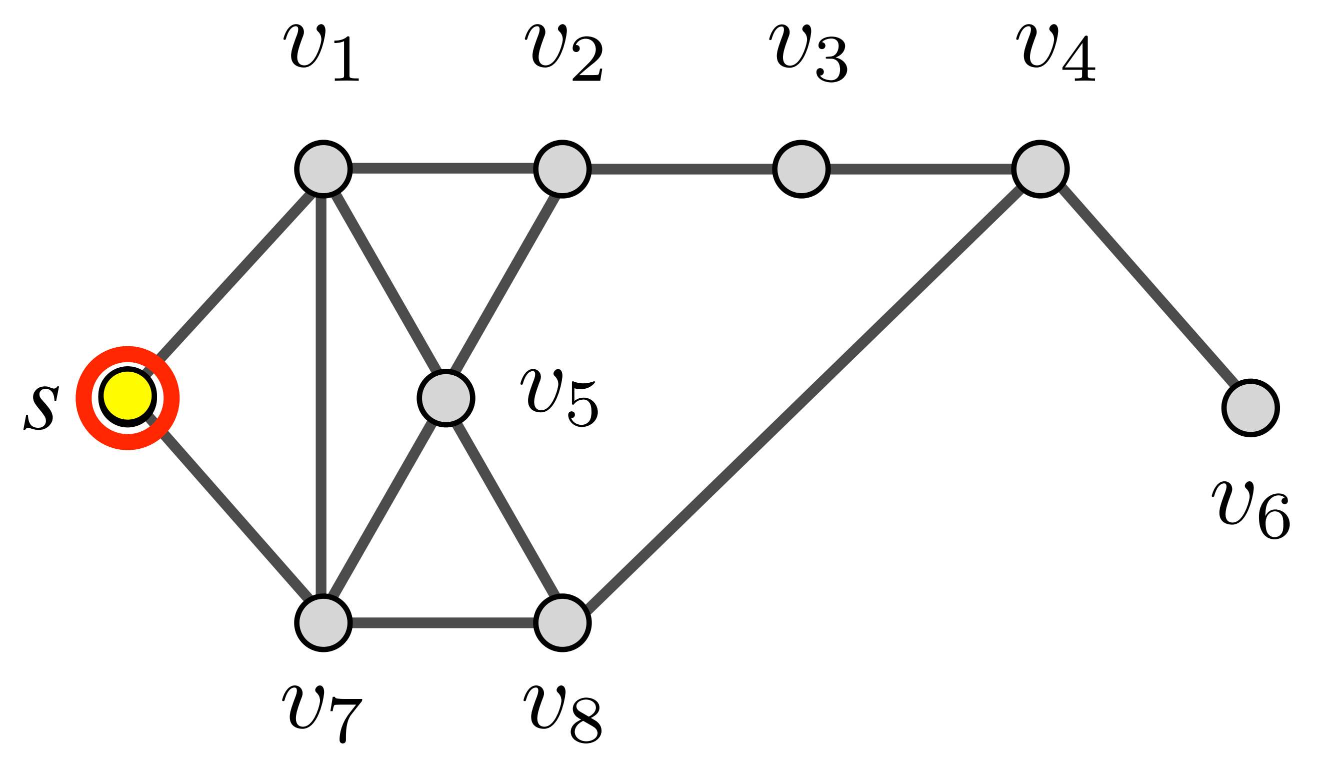
Algorithmus 3.7

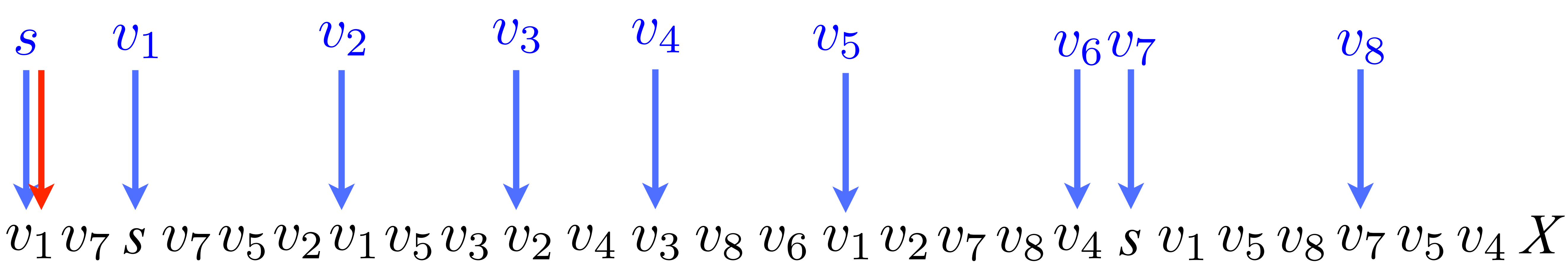
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

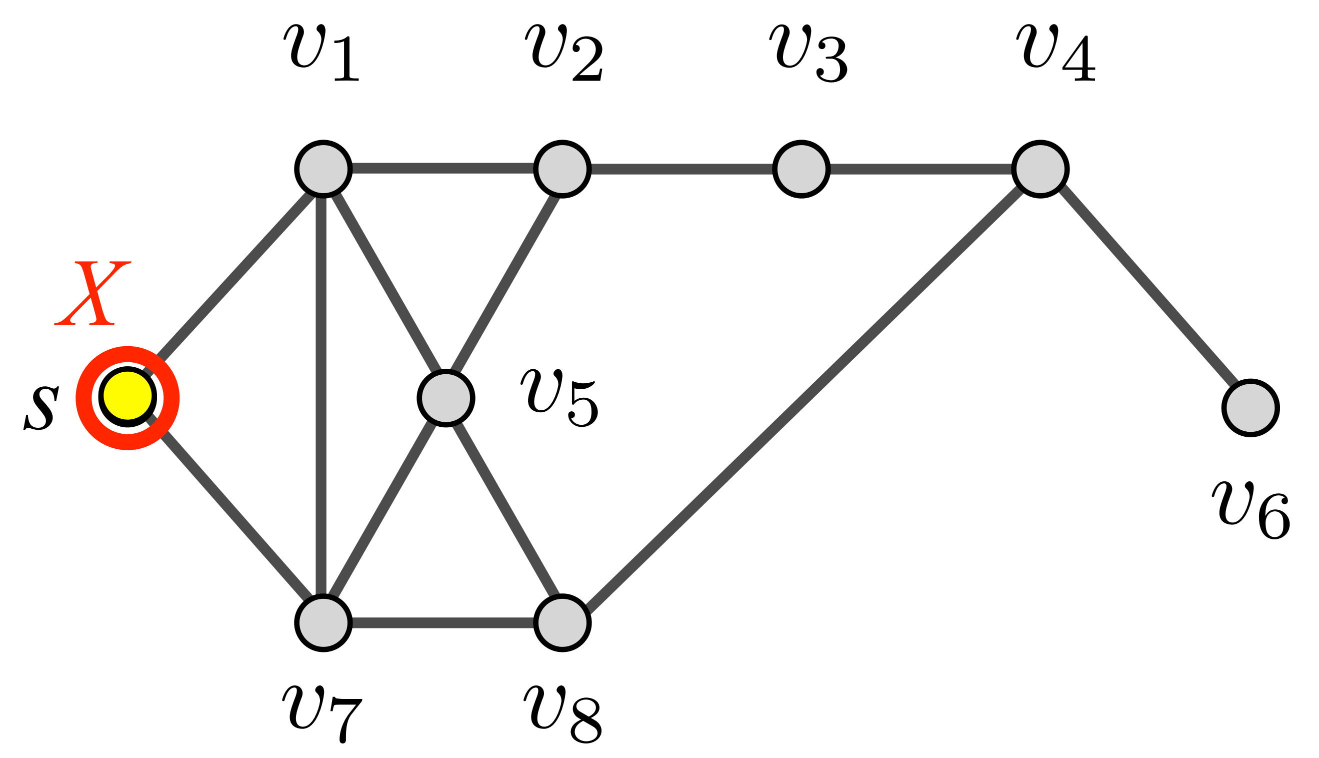




Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$v_1\ v_7\ s\ v_7\ v_5\ v_2\ v_1\ v_5\ v_3\ v_2\ v_4\ v_3\ v_8\ v_6\ v_1\ v_2\ v_7\ v_8\ v_4\ s\ v_1\ v_5\ v_8\ v_7\ v_5\ v_4\ X$

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

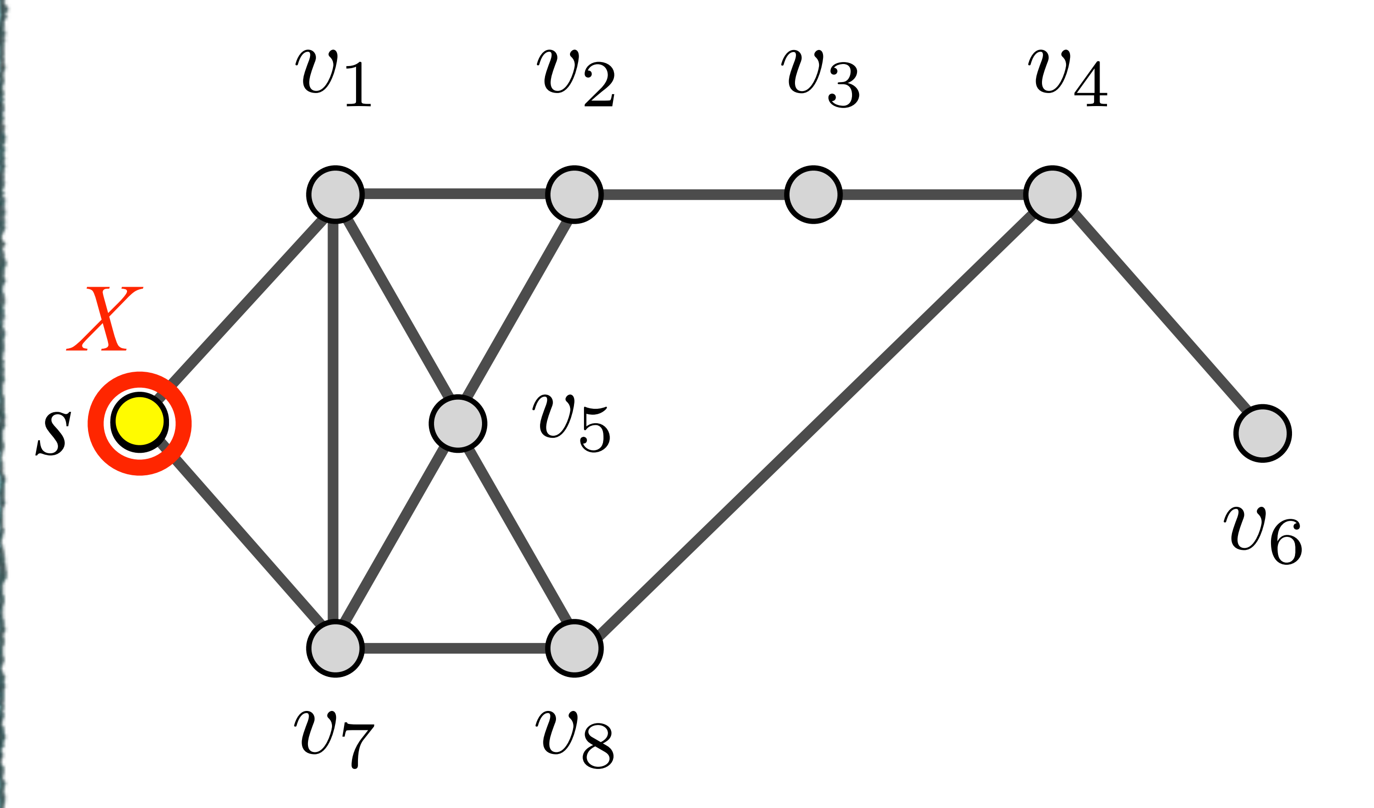
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

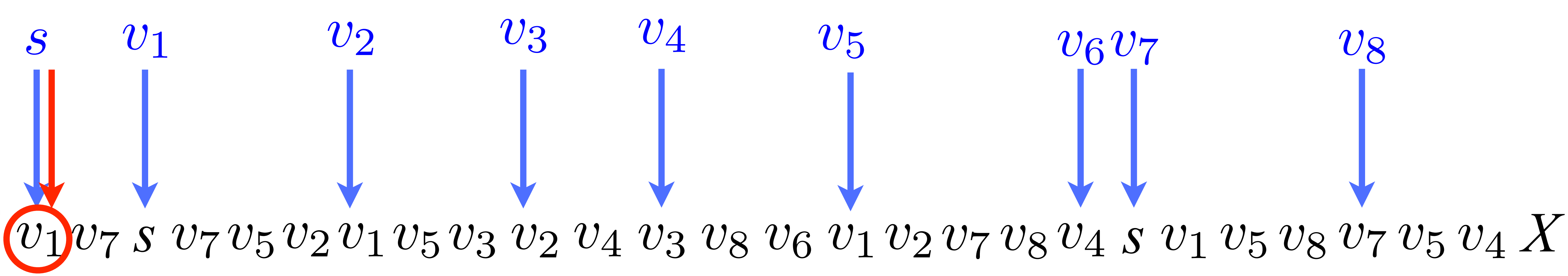
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





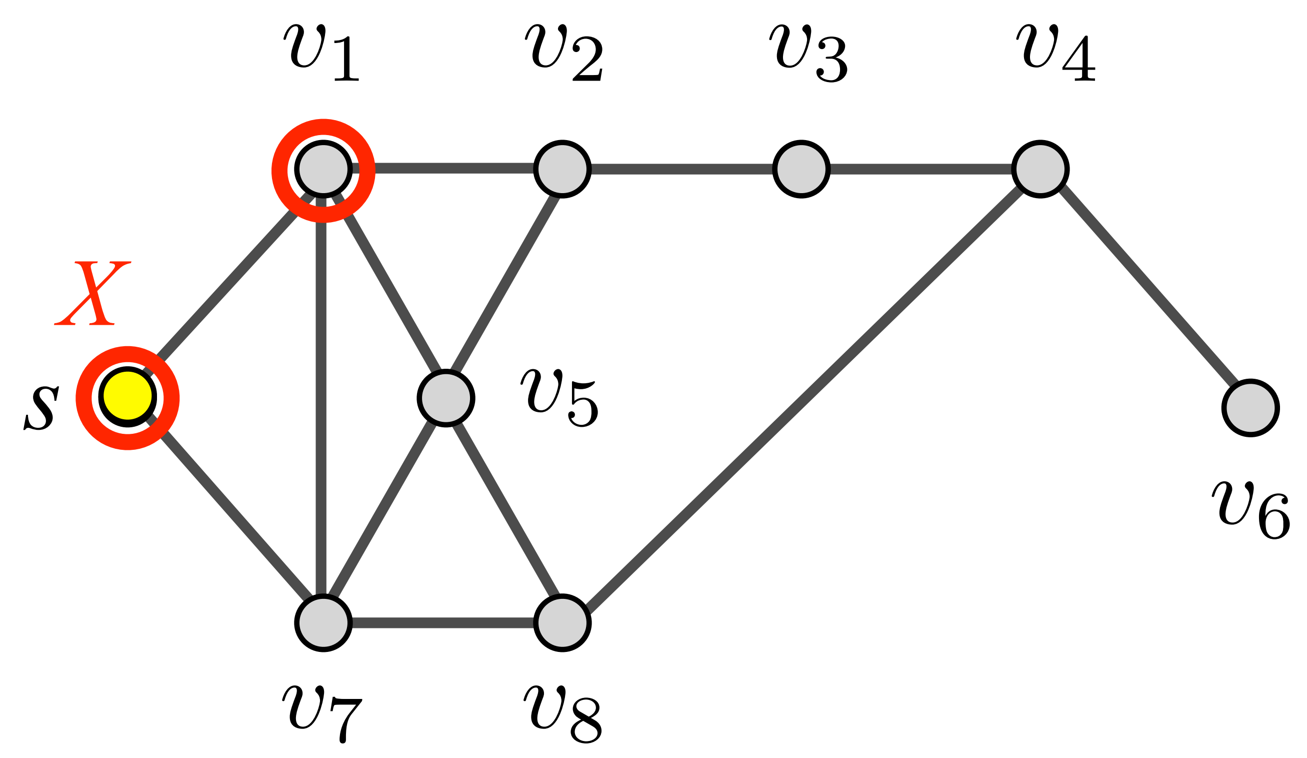
Algorithmus 3.7

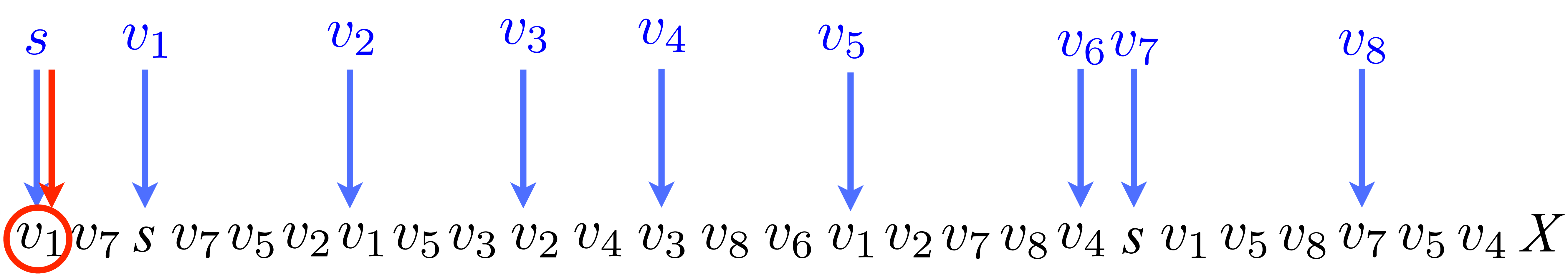
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP

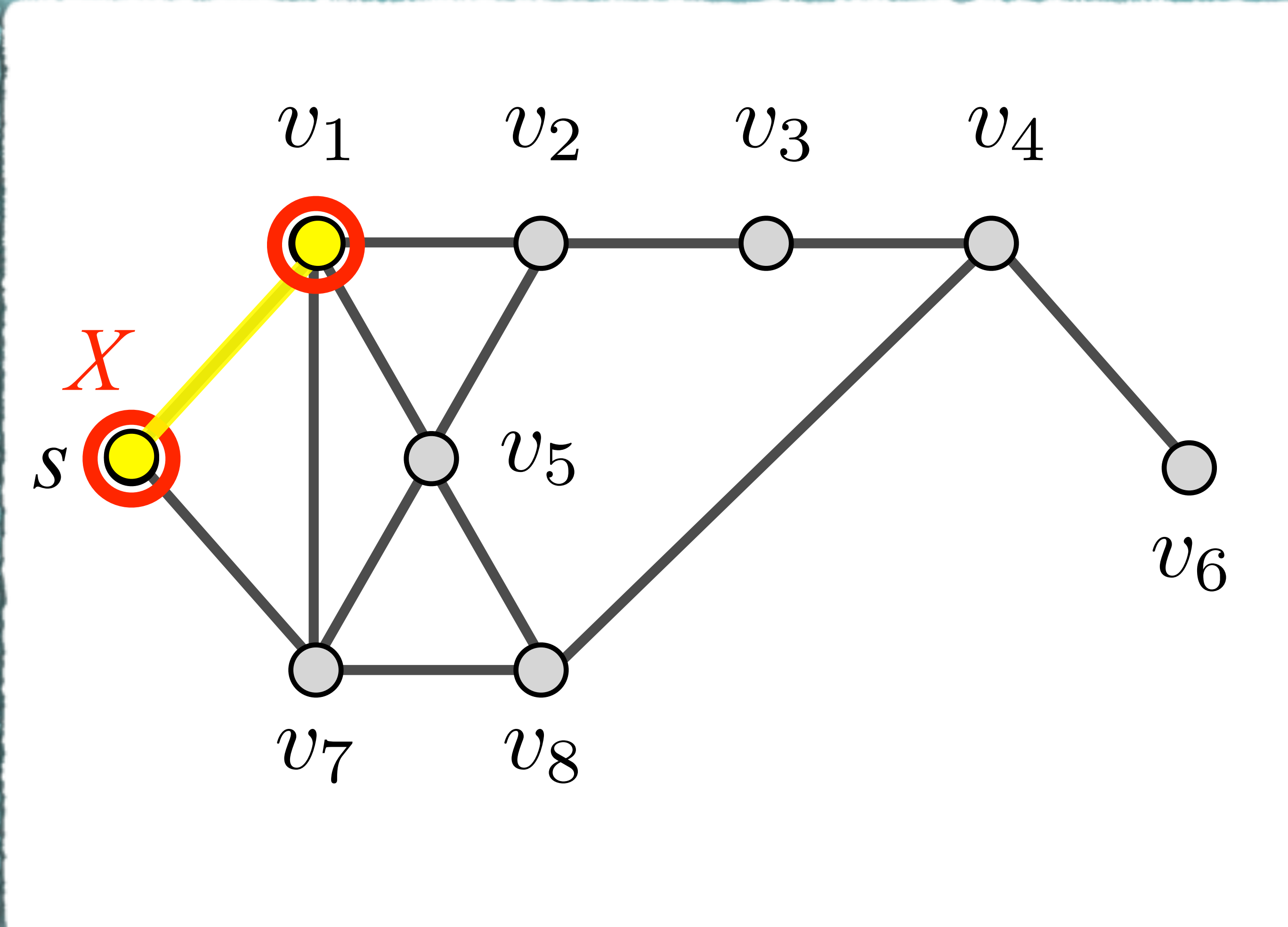


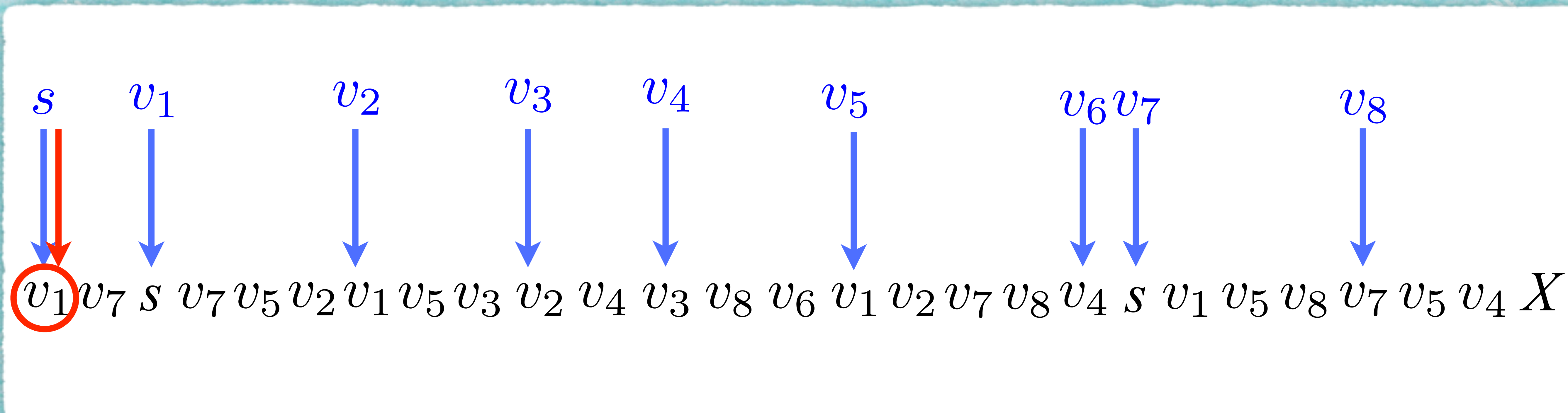


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
   }
3. STOP





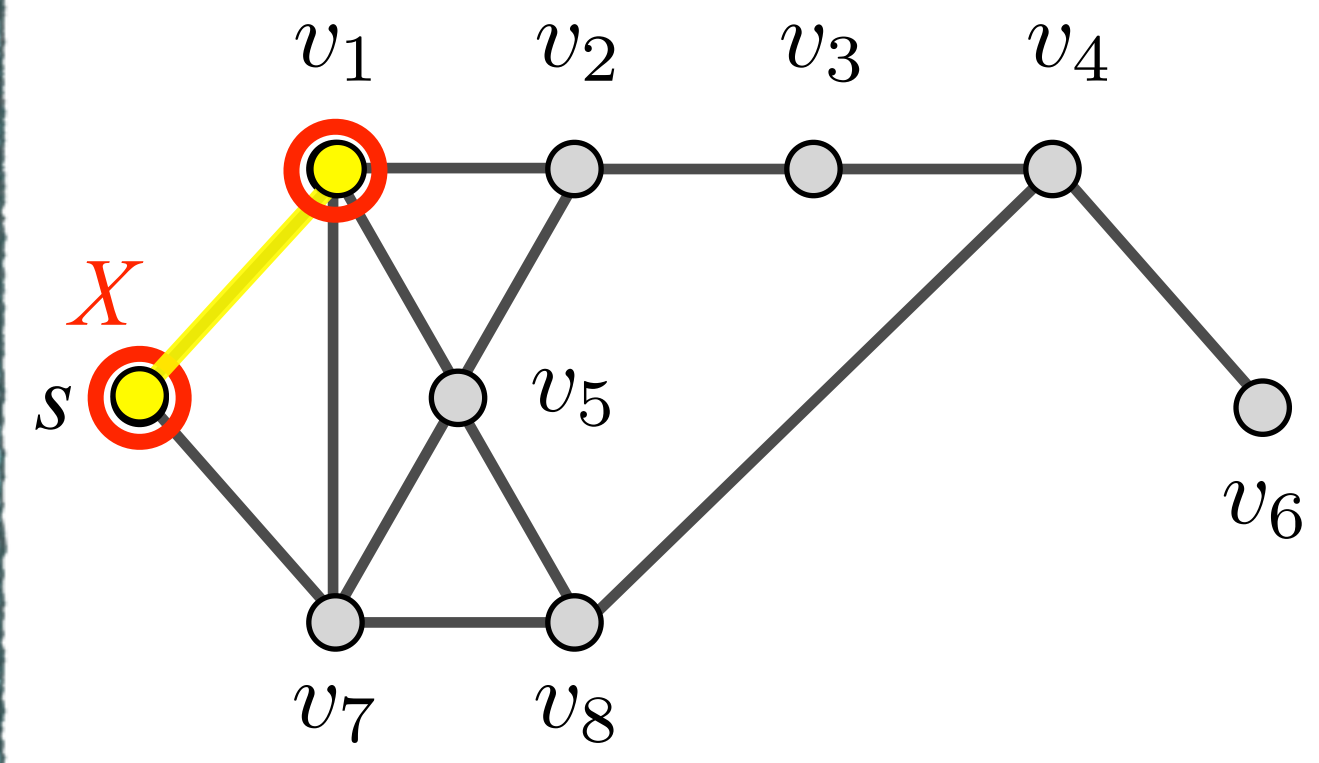
Algorithmus 3.7

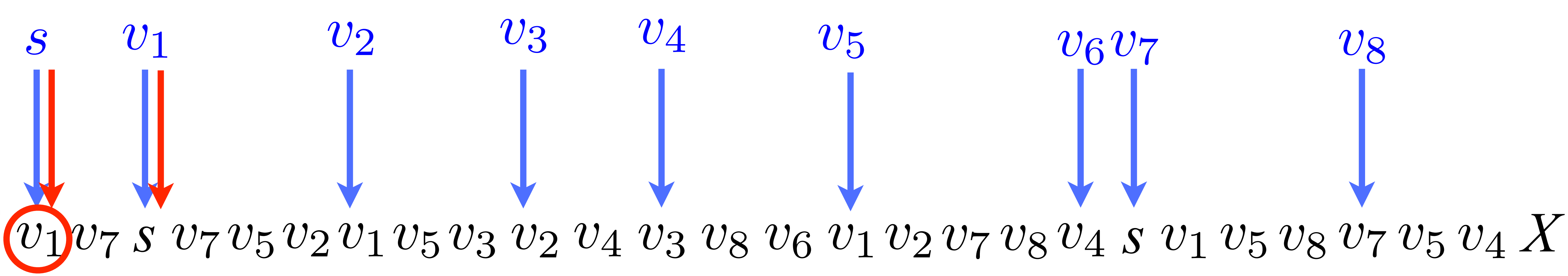
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

```
1.  Sei R:={s}, Y:={s}, T:=∅
2.  WHILE (R≠∅) DO {
      2.1.  Wähle v∈R
      2.2.  IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
            2.2.1.  R:=R\{v}
      2.3.  ELSE {
            2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
            2.3.2.  Setze R := R u {w}, Y :=  Y u {w}, T := T u {e}
            }
    }
3.  STOP
```



Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

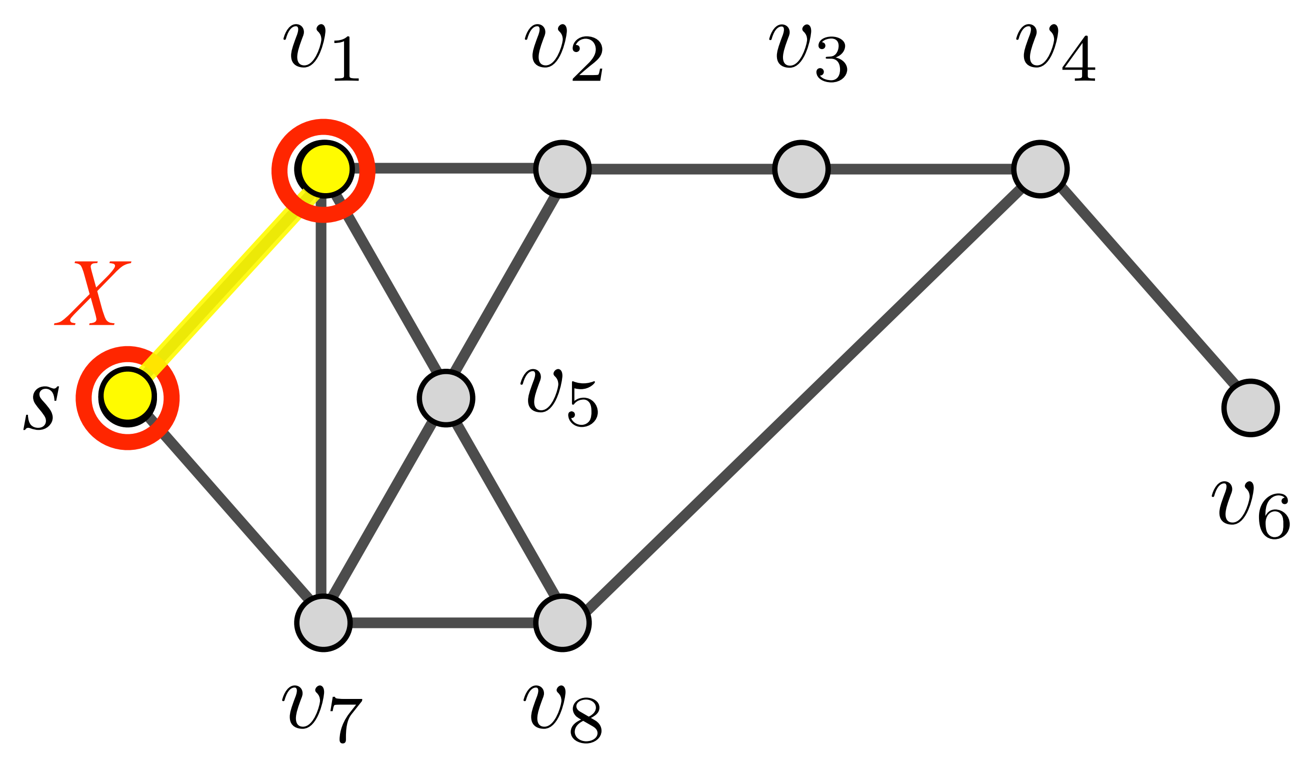
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

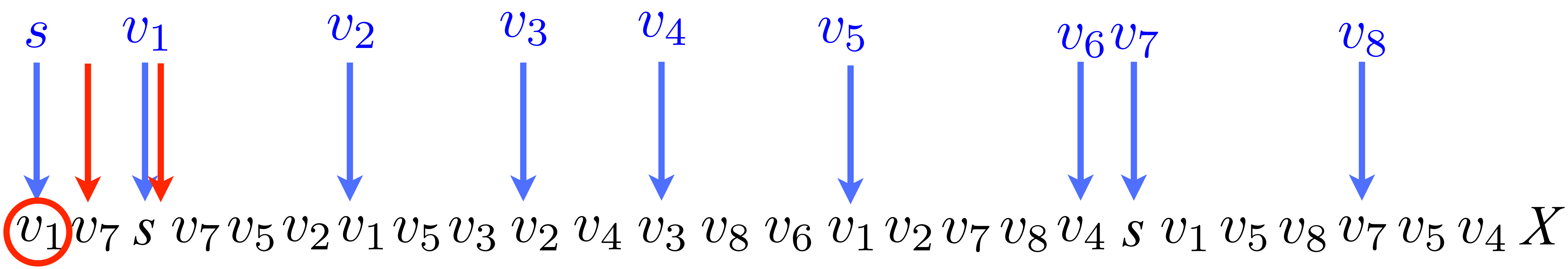
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP

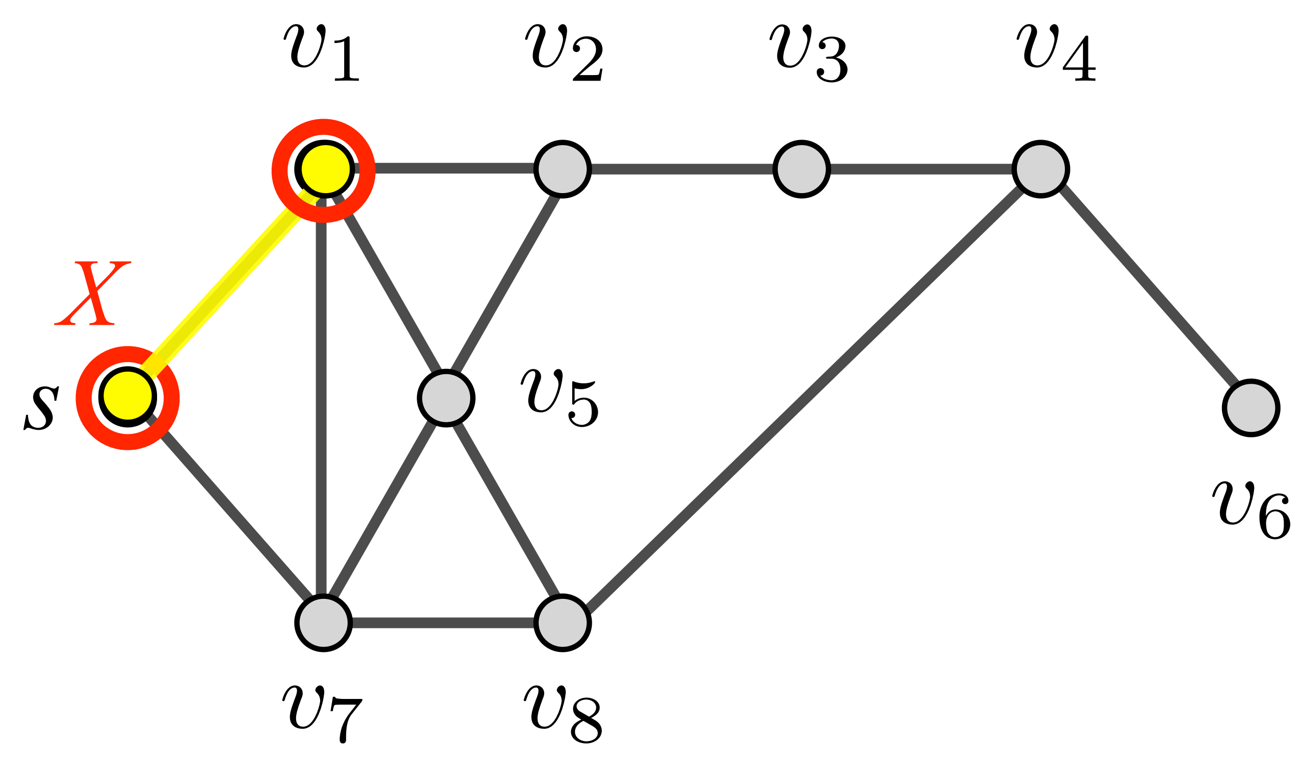


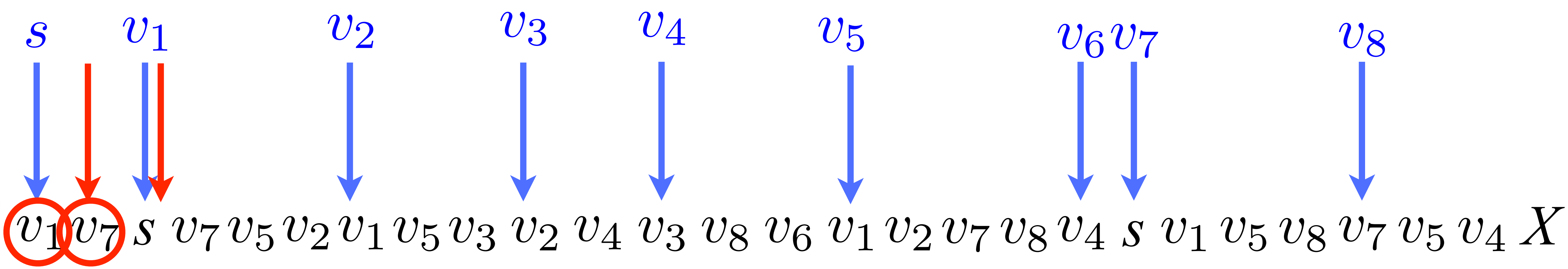


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





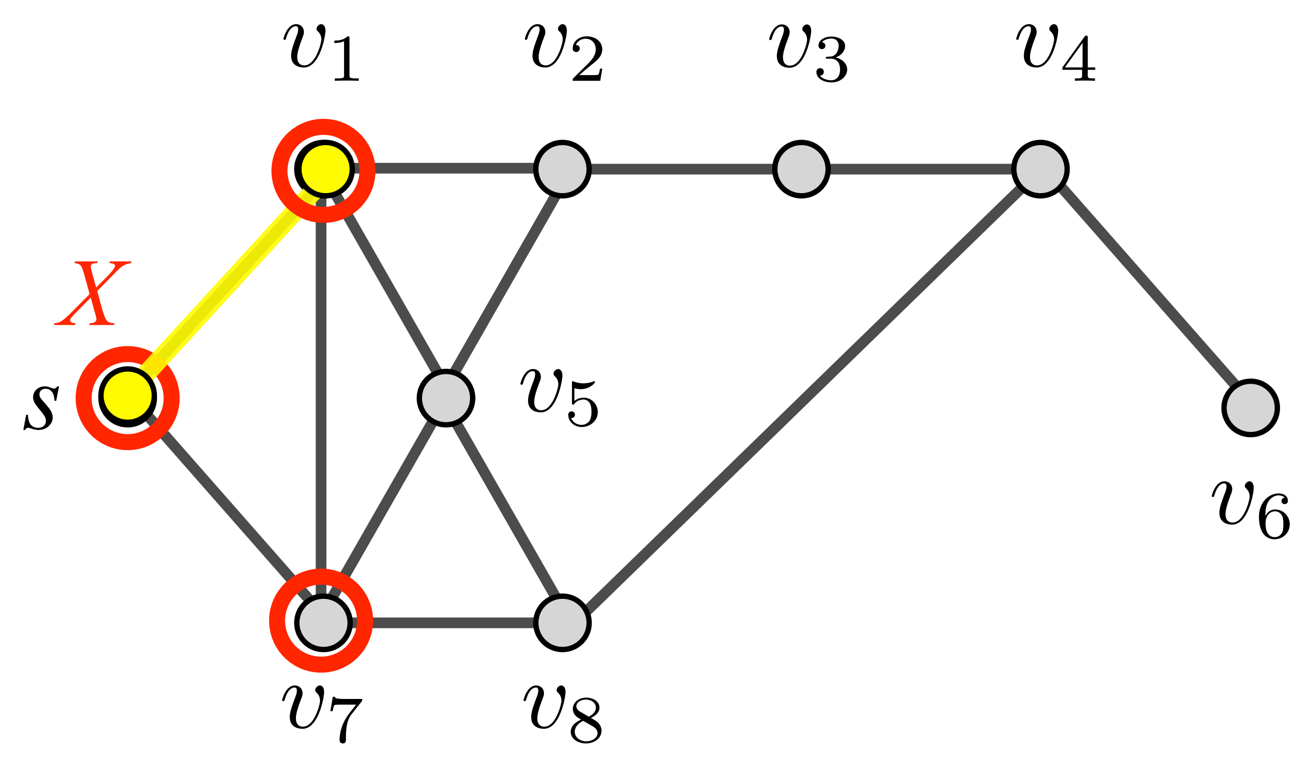
Algorithmus 3.7

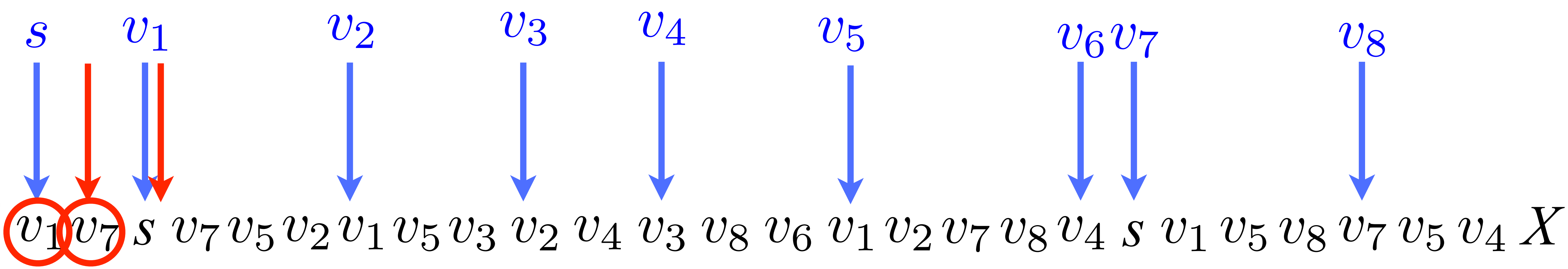
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP

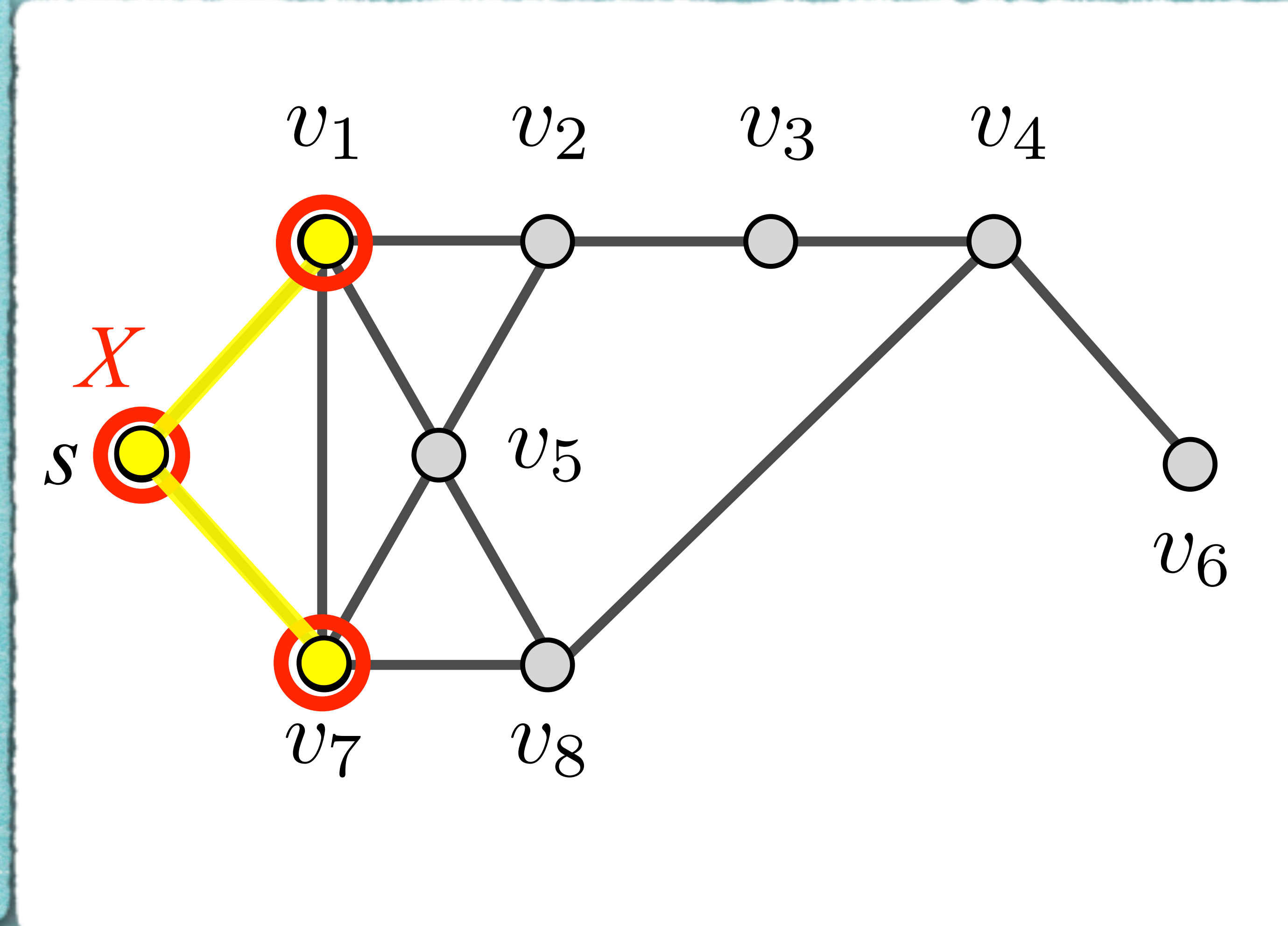




Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }
    - }
3. STOP



s v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

$$v_1\, v_7\, s\, v_7\, v_5\, v_2\, v_1\, v_5\, v_3\, v_2\, v_4\, v_3\, v_8\, v_6\, v_1\, v_2\, v_7\, v_8\, v_4\, s\, v_1\, v_5\, v_8\, v_7\, v_5\, v_4\, X$$

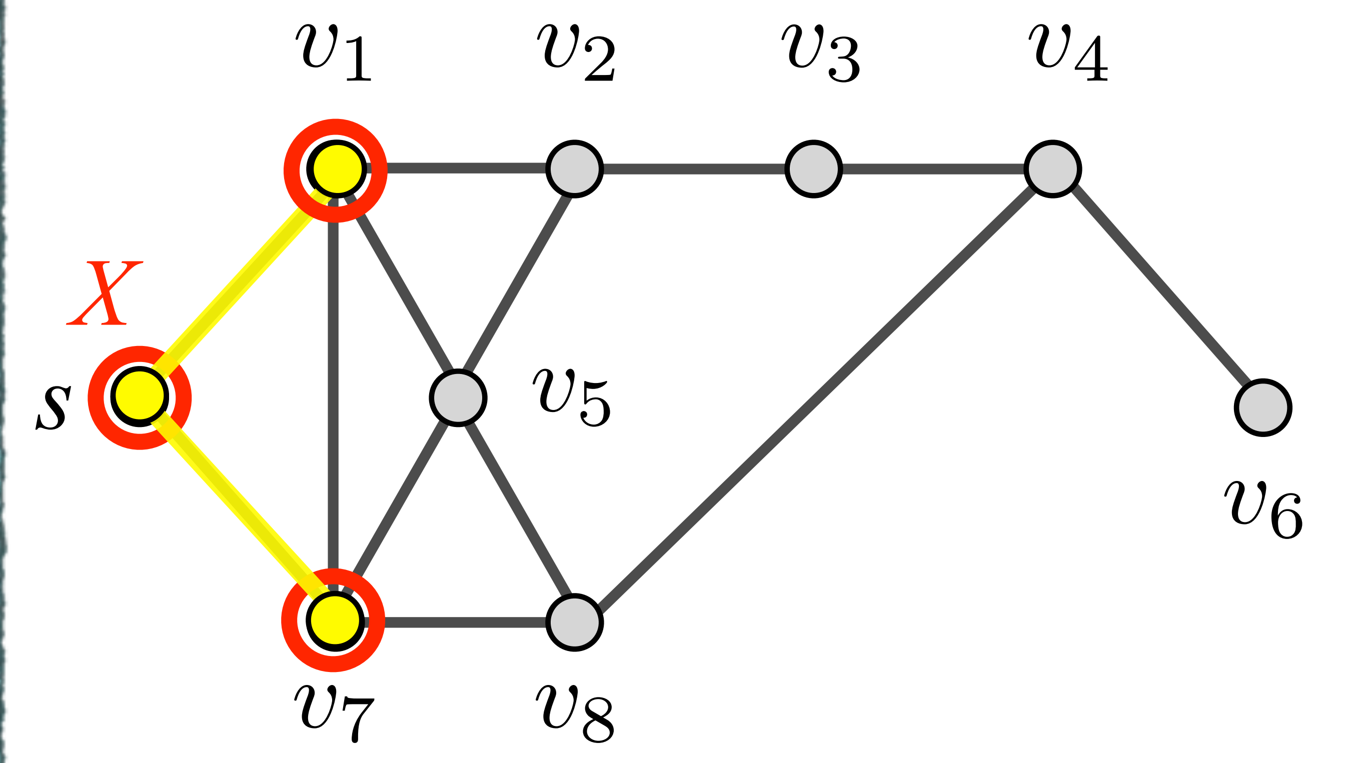
Algorithmus 3.7

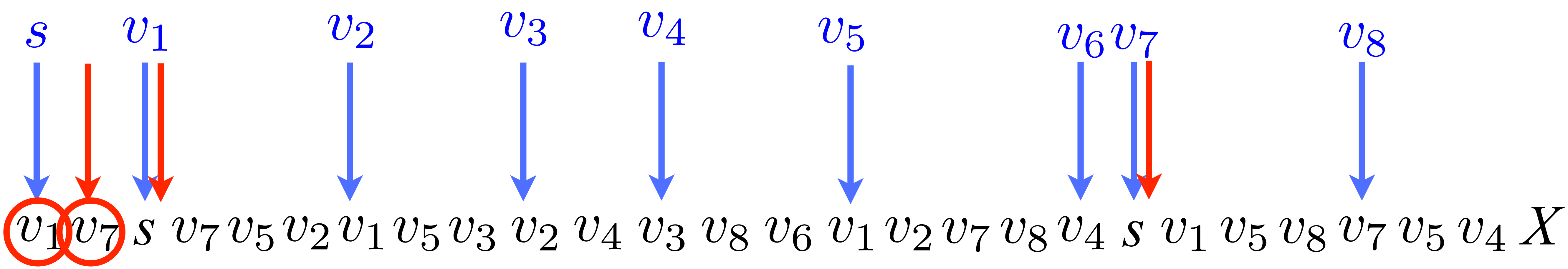
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

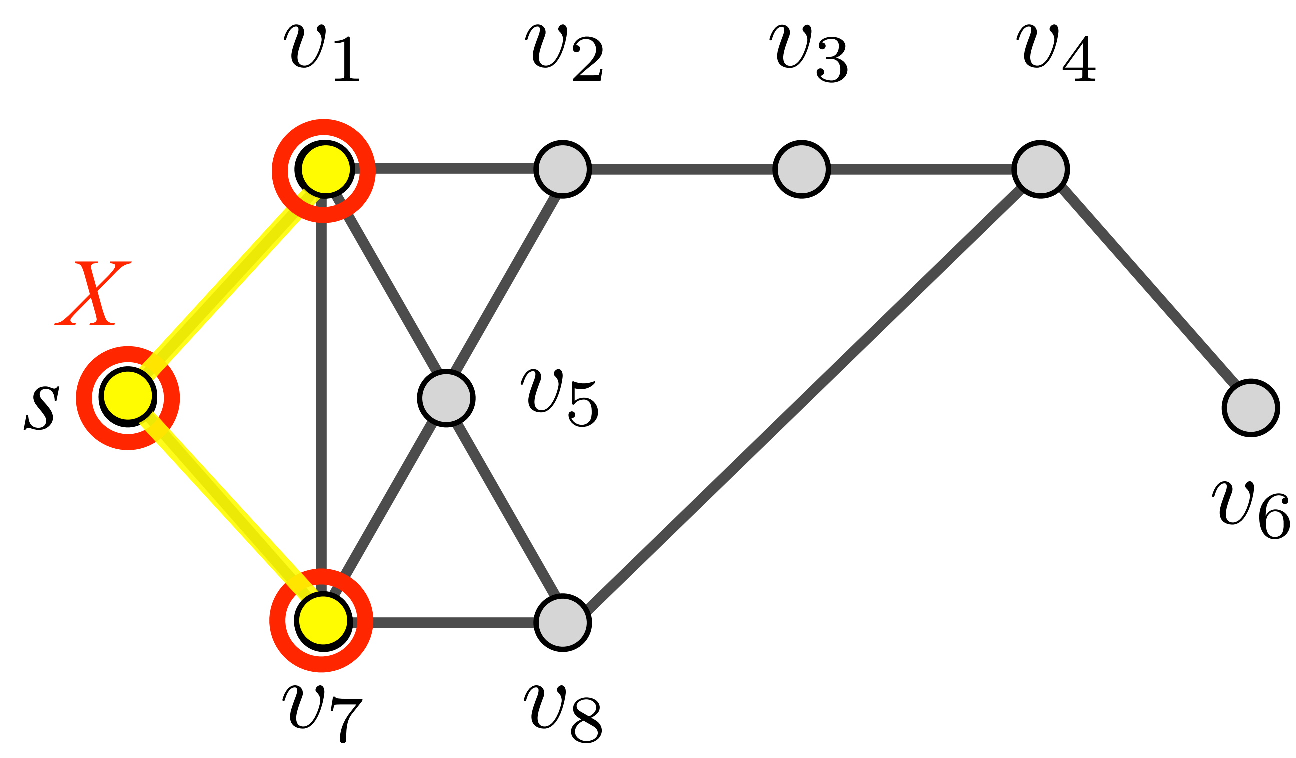
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

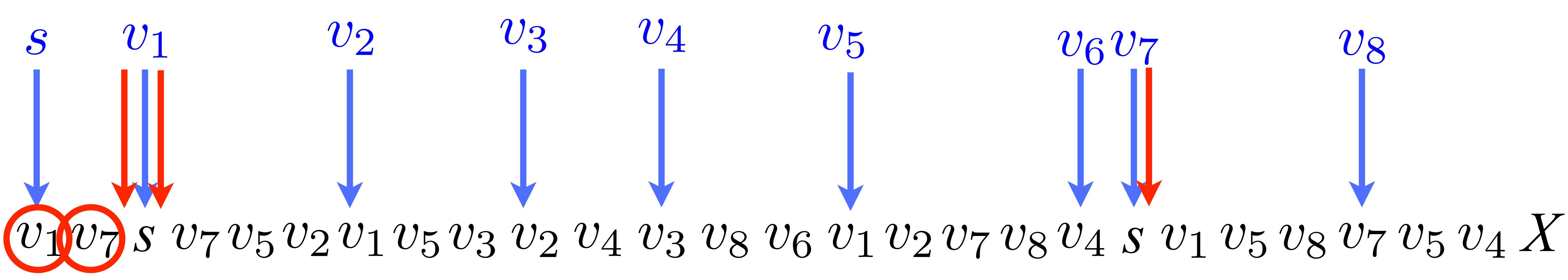
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈R
    - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

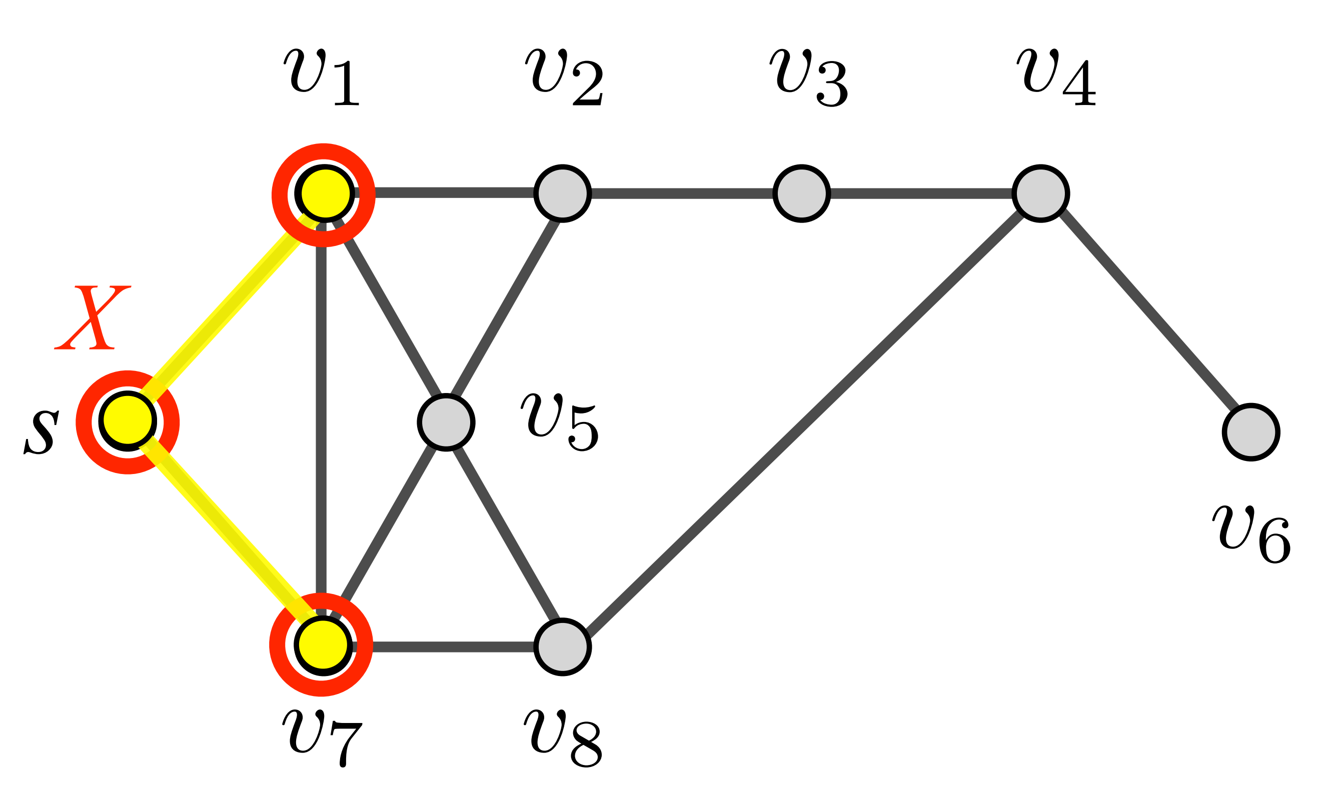
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

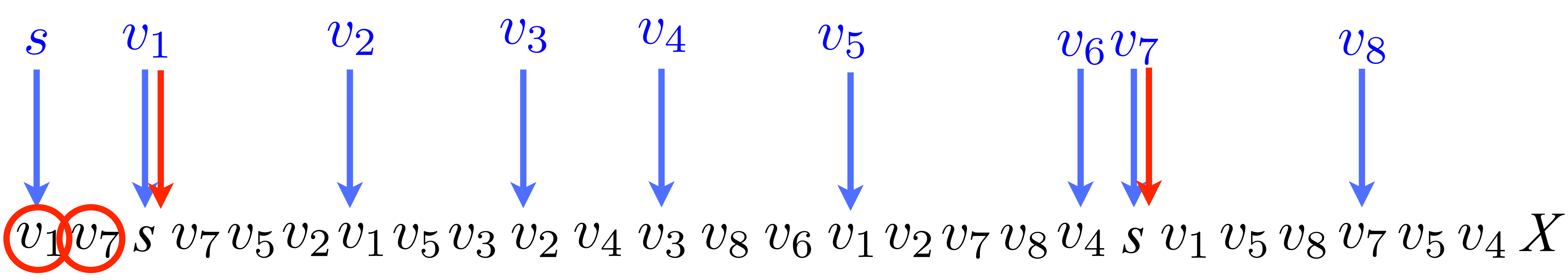
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP



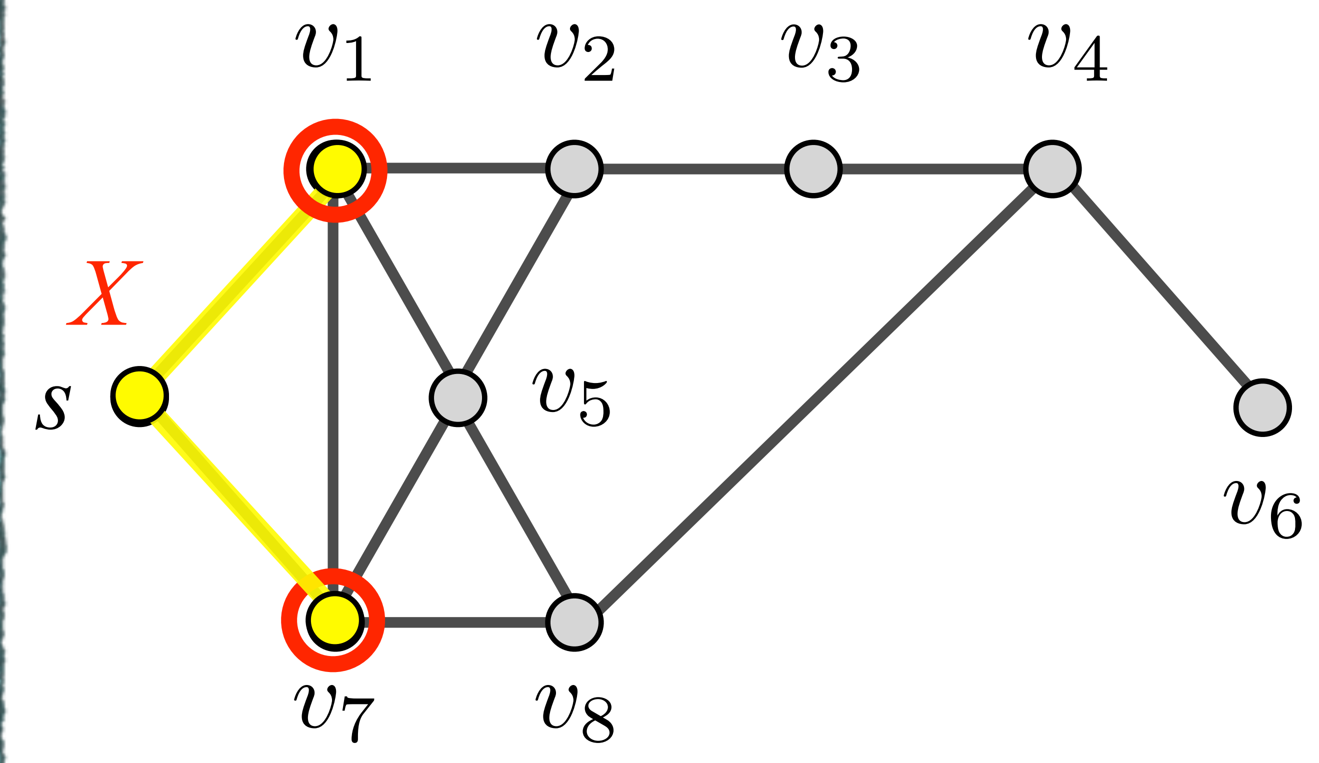


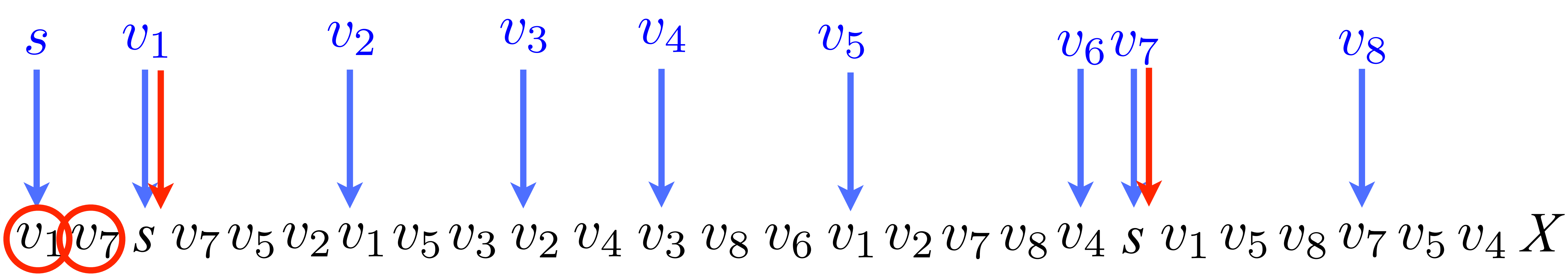
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }

    }
3. STOP





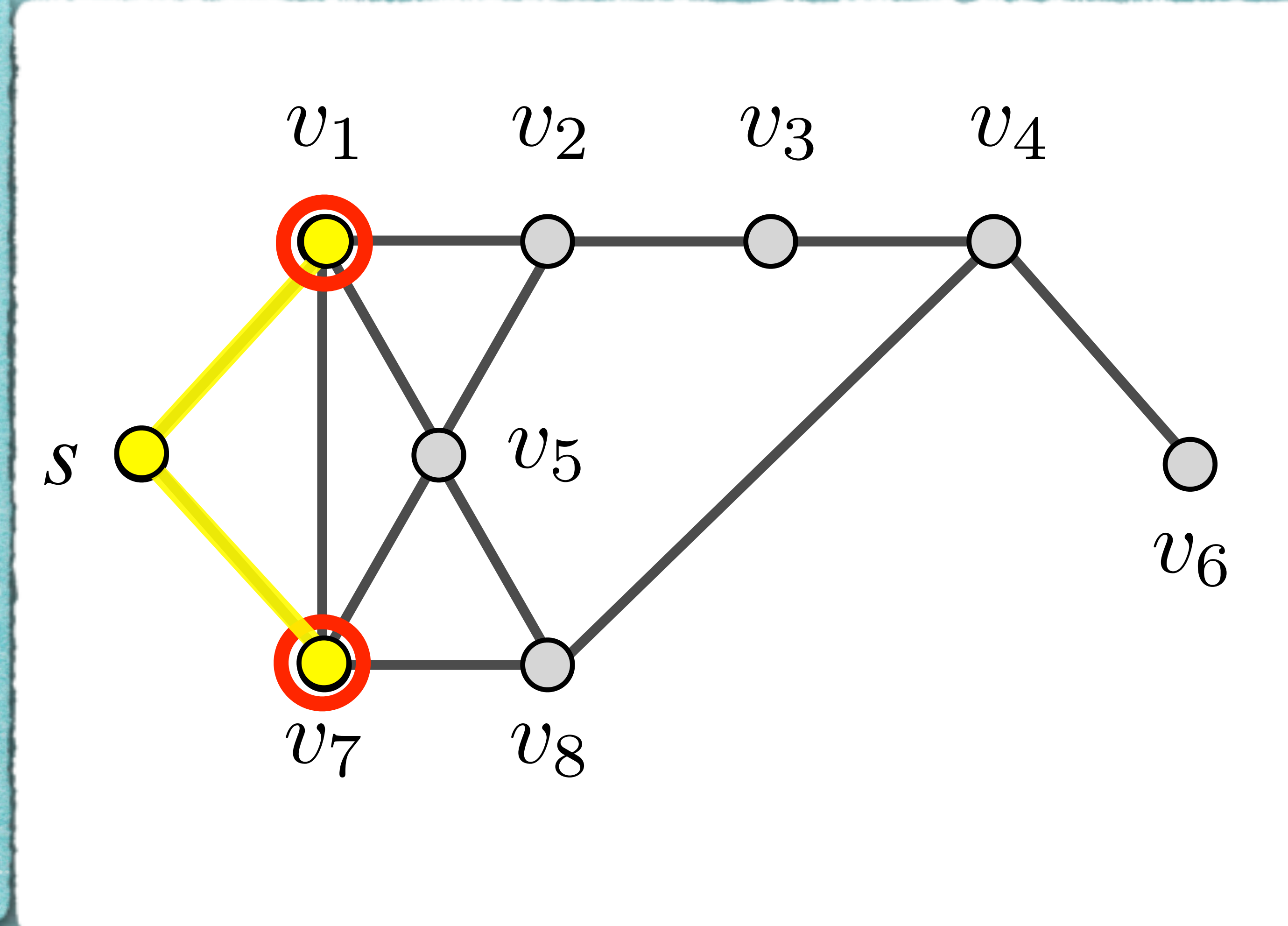
Algorithmus 3.7

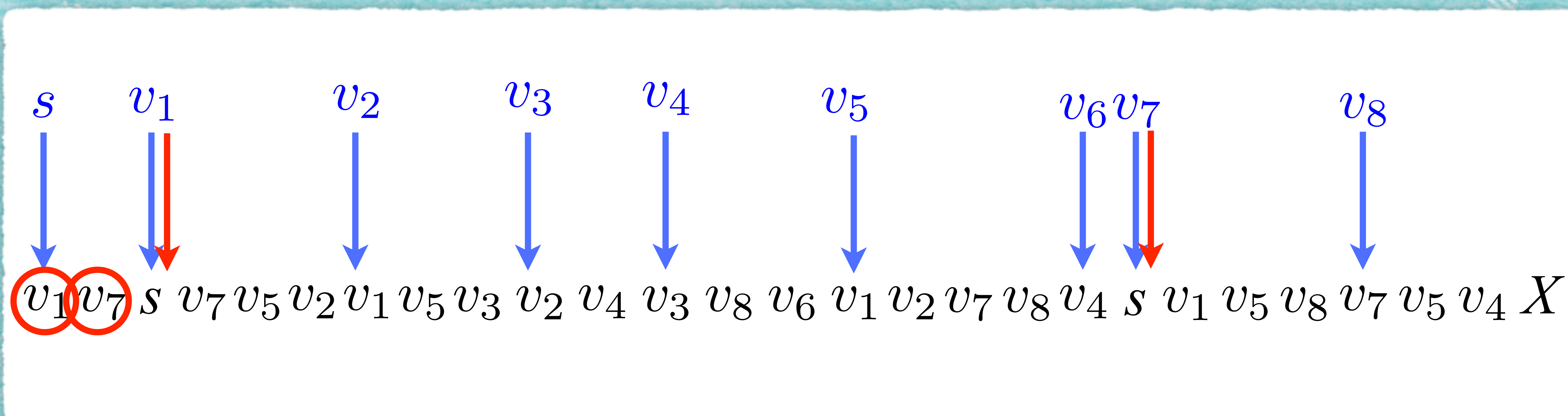
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

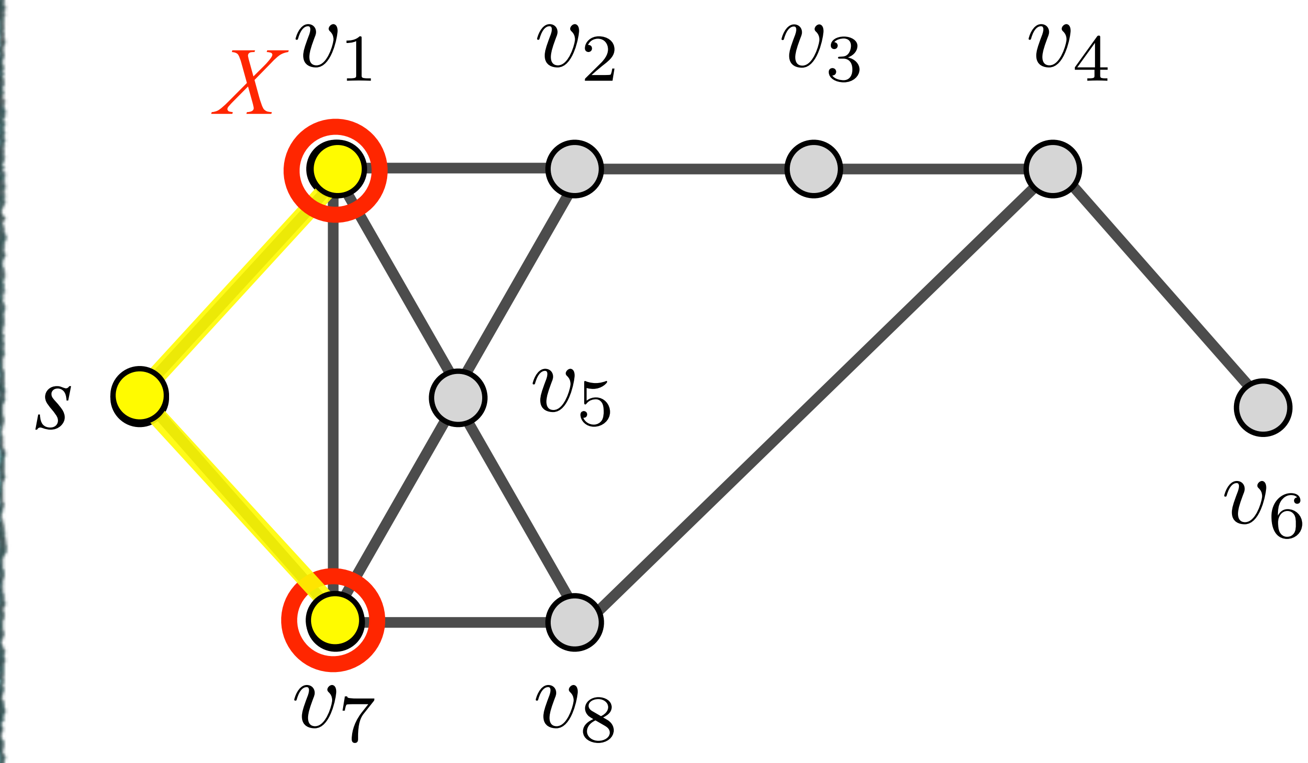
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

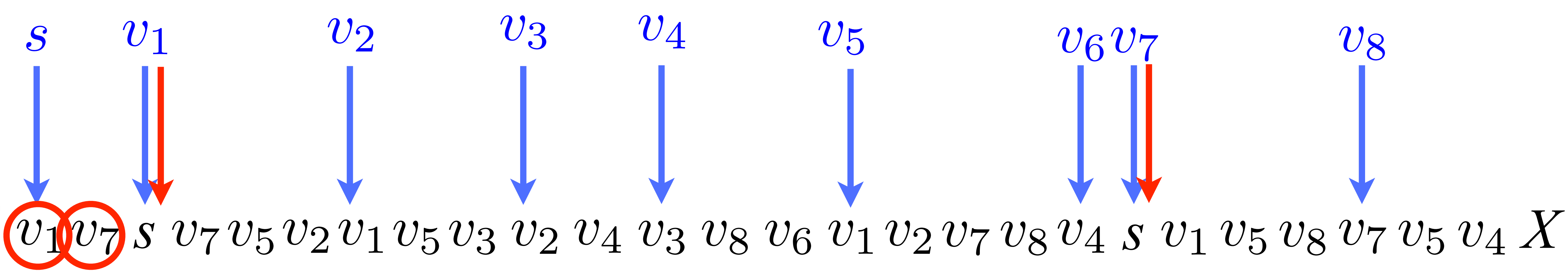
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

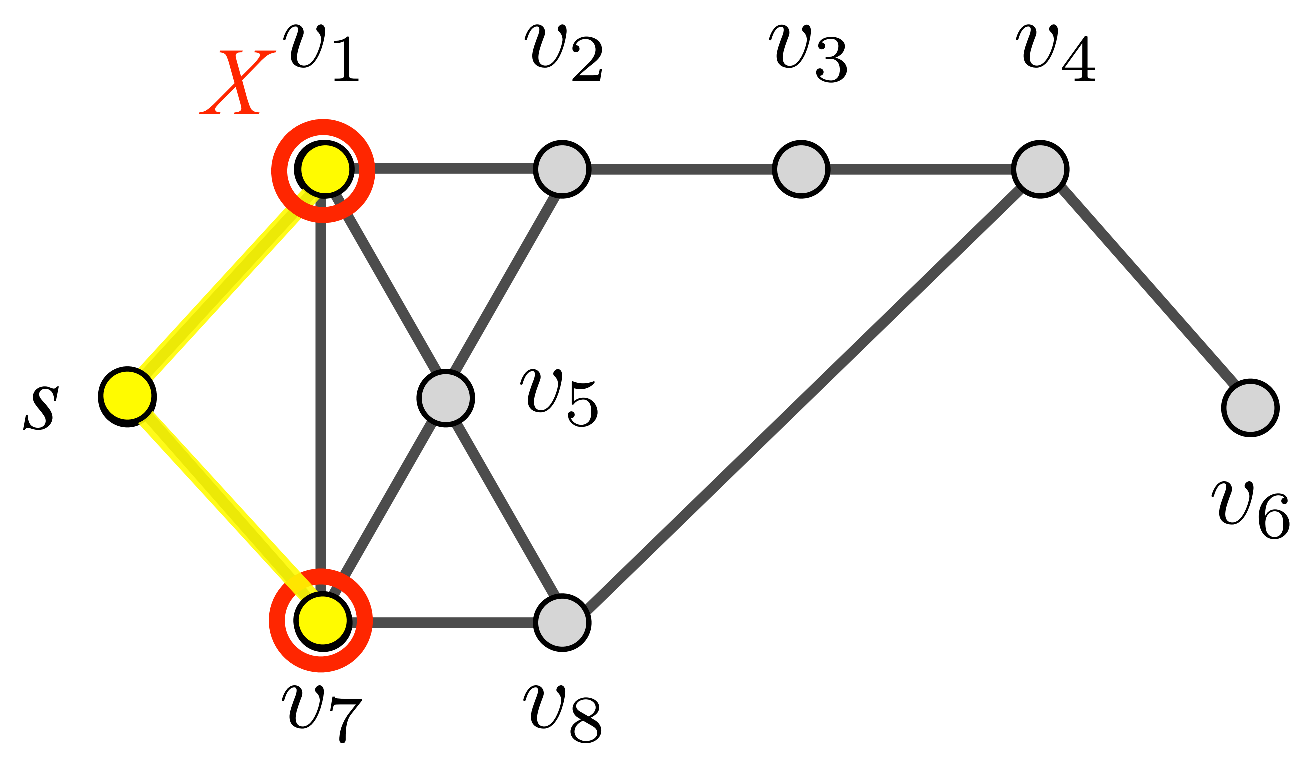
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

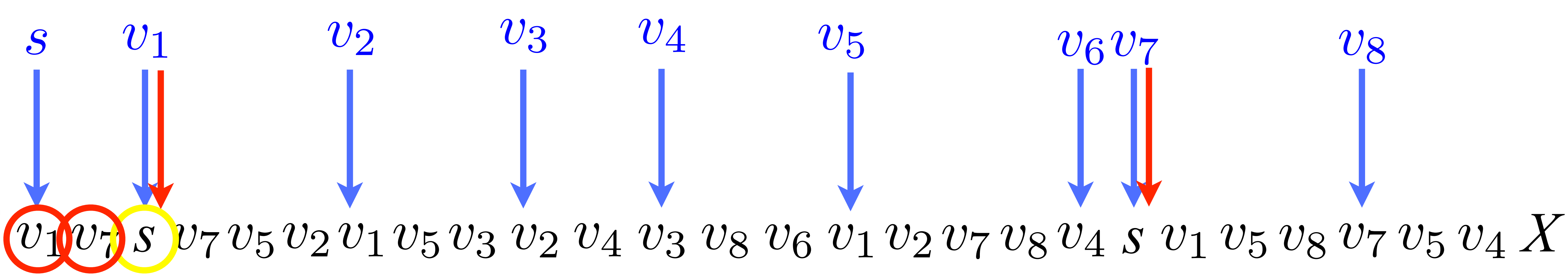
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP

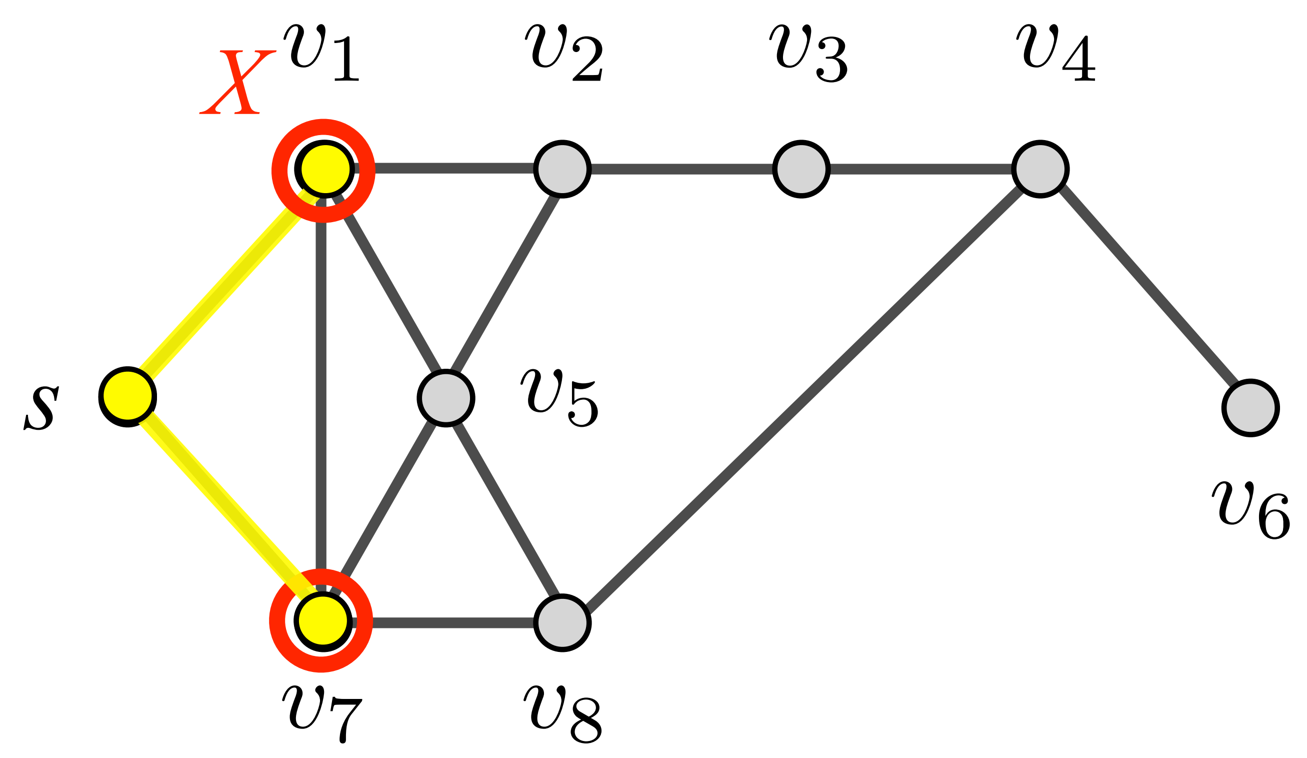


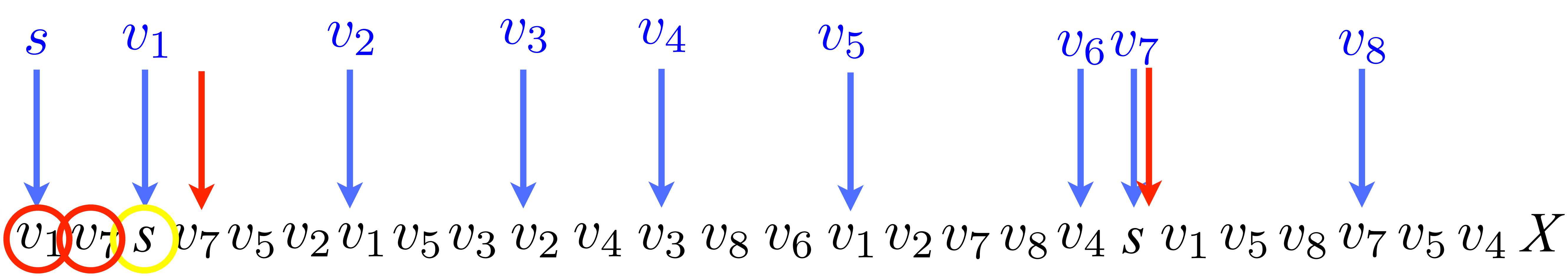


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈ R
   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
      2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
      }
   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

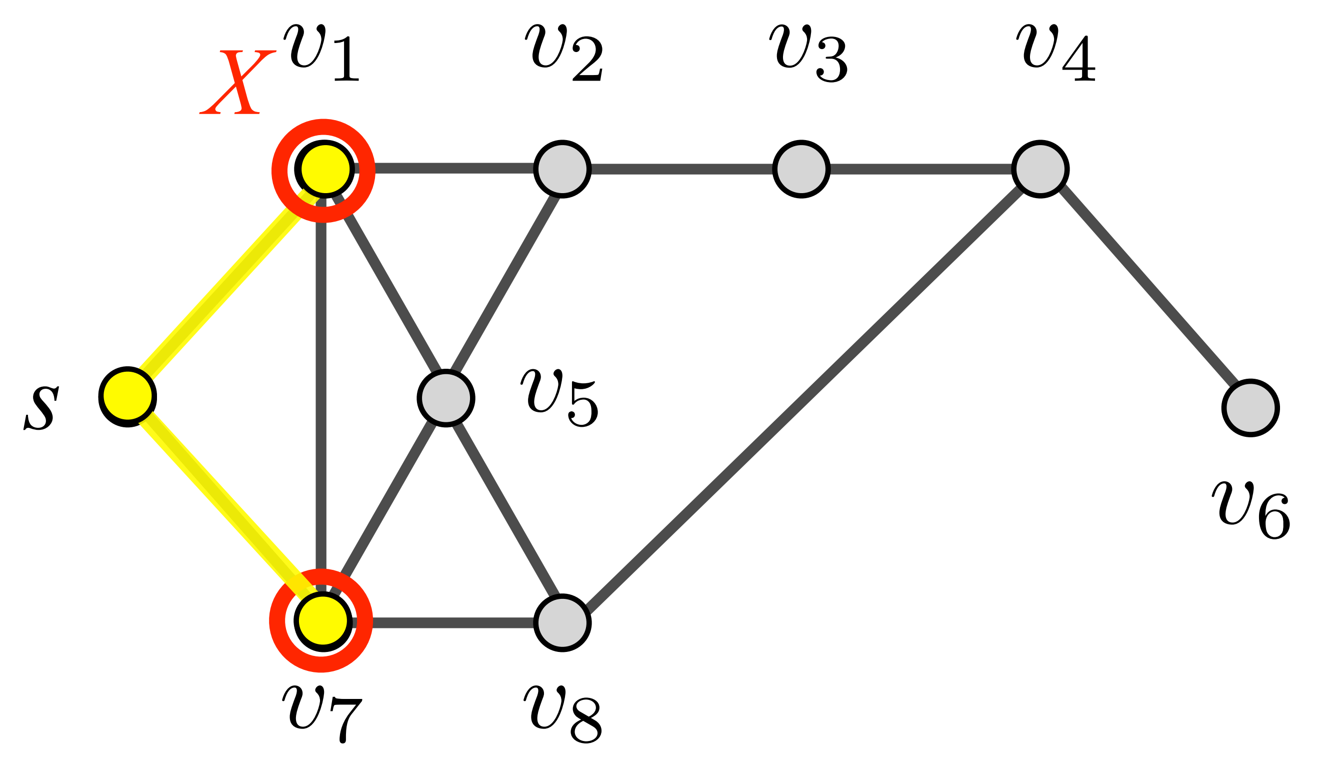
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

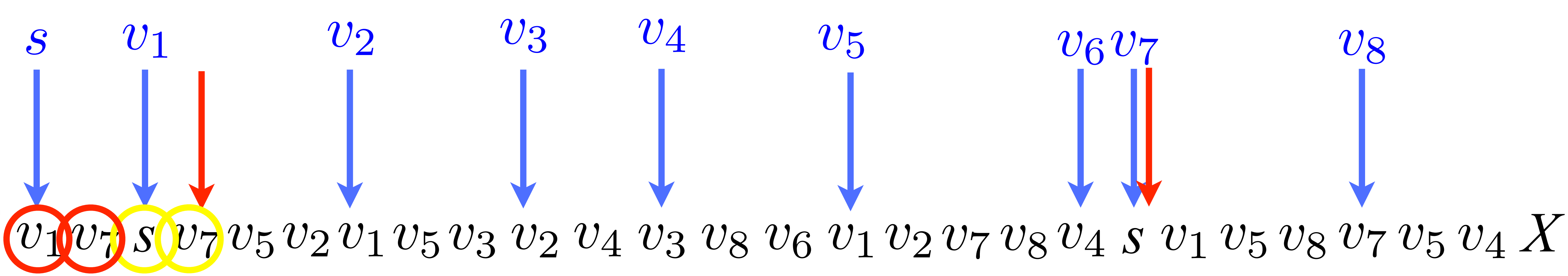
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

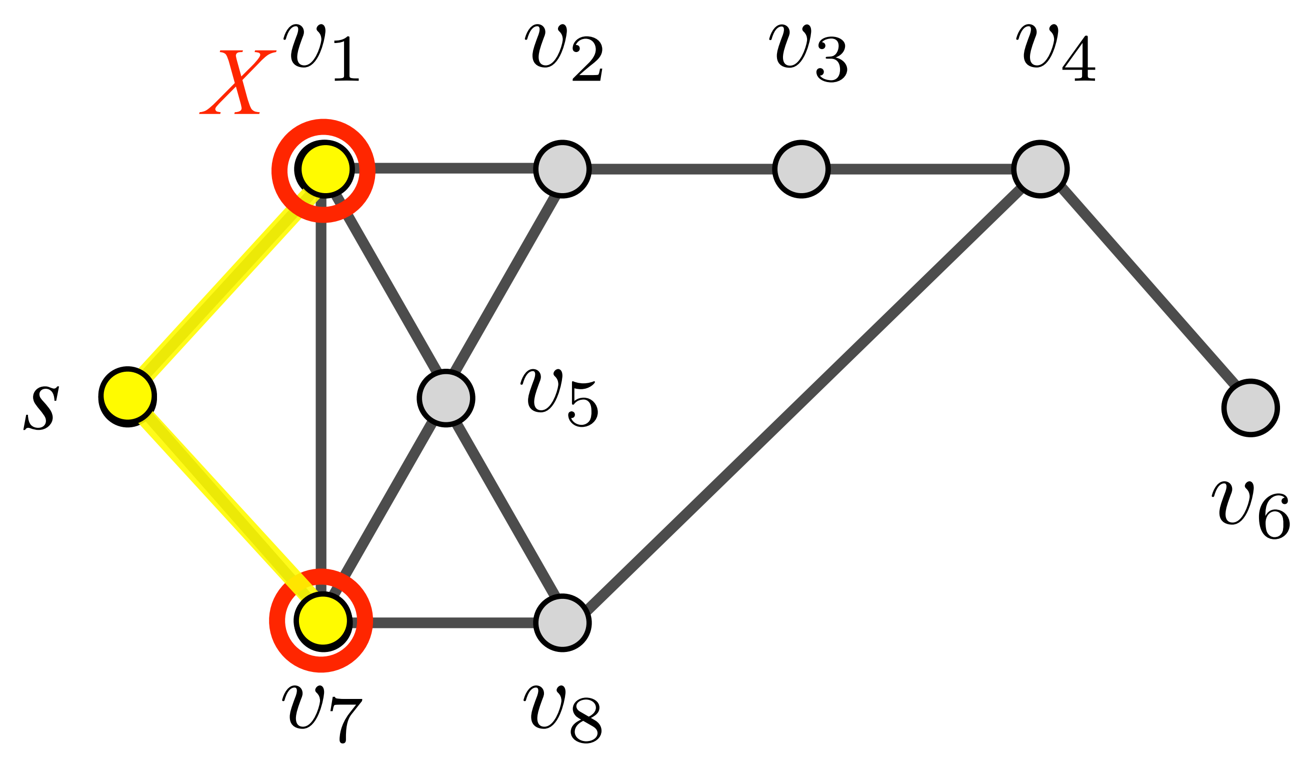
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

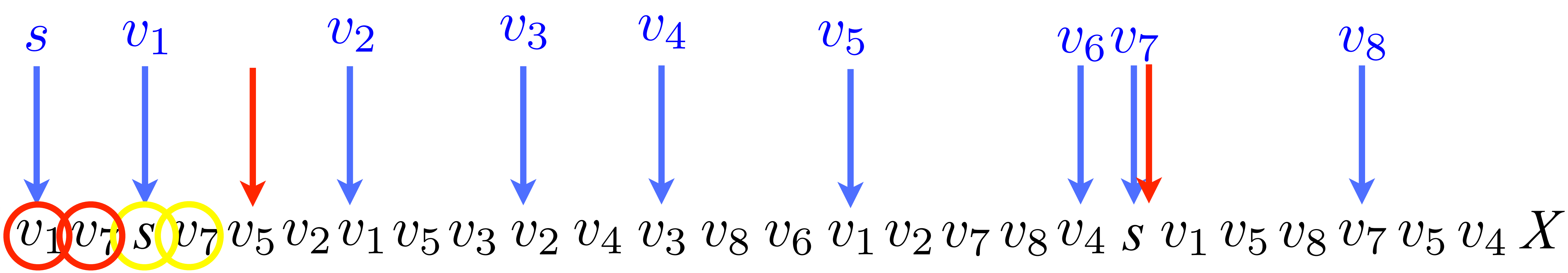
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP

2





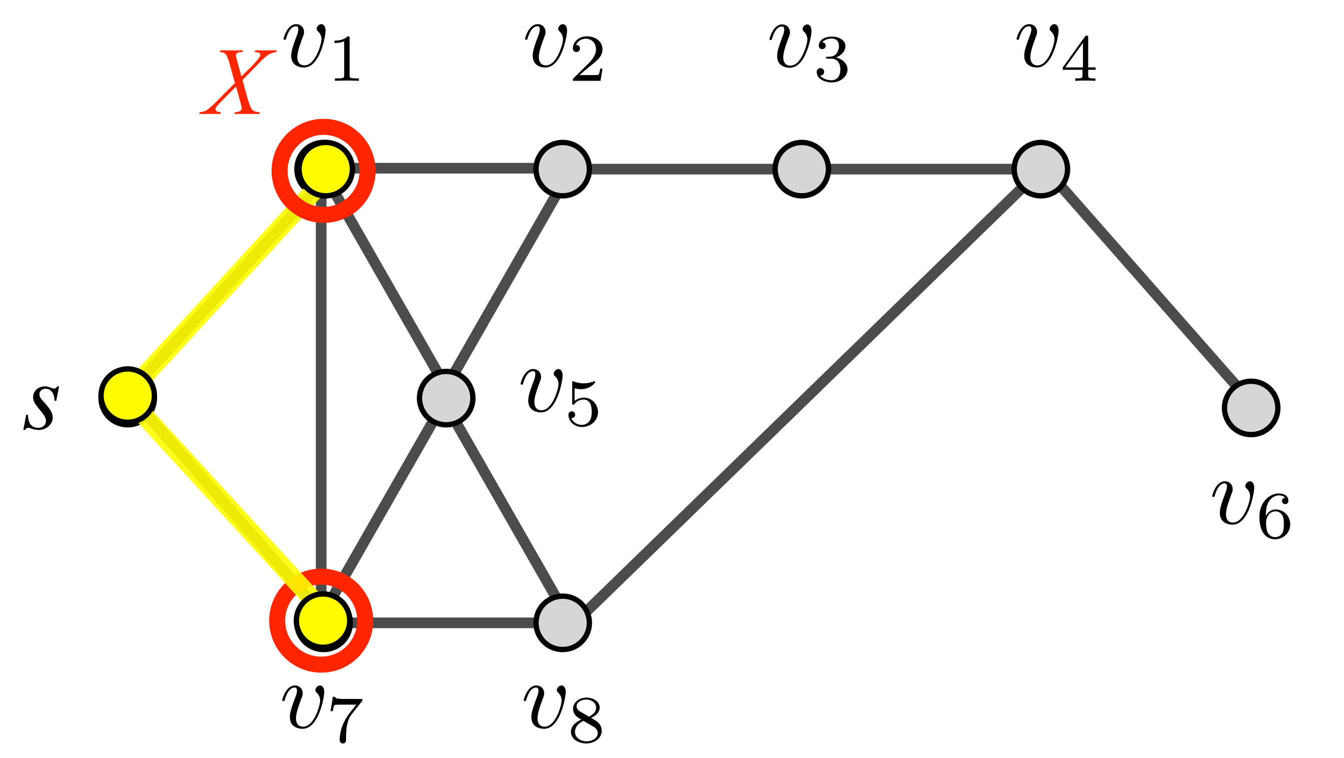
Algorithmus 3.7

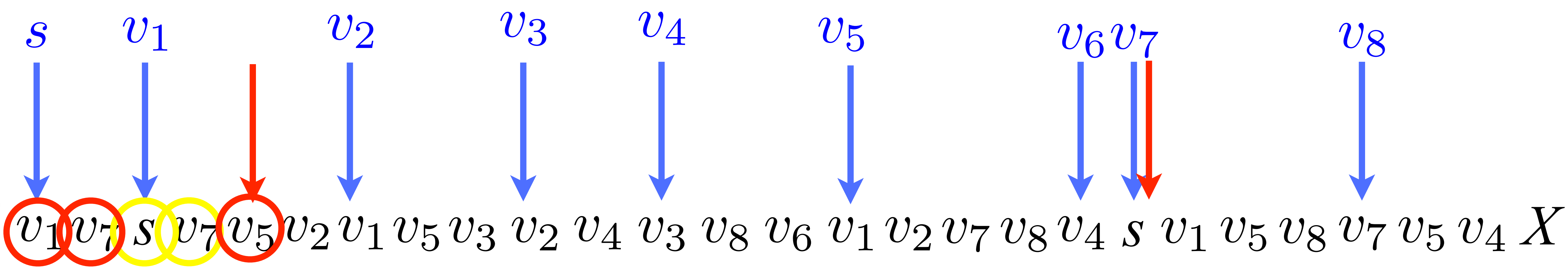
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





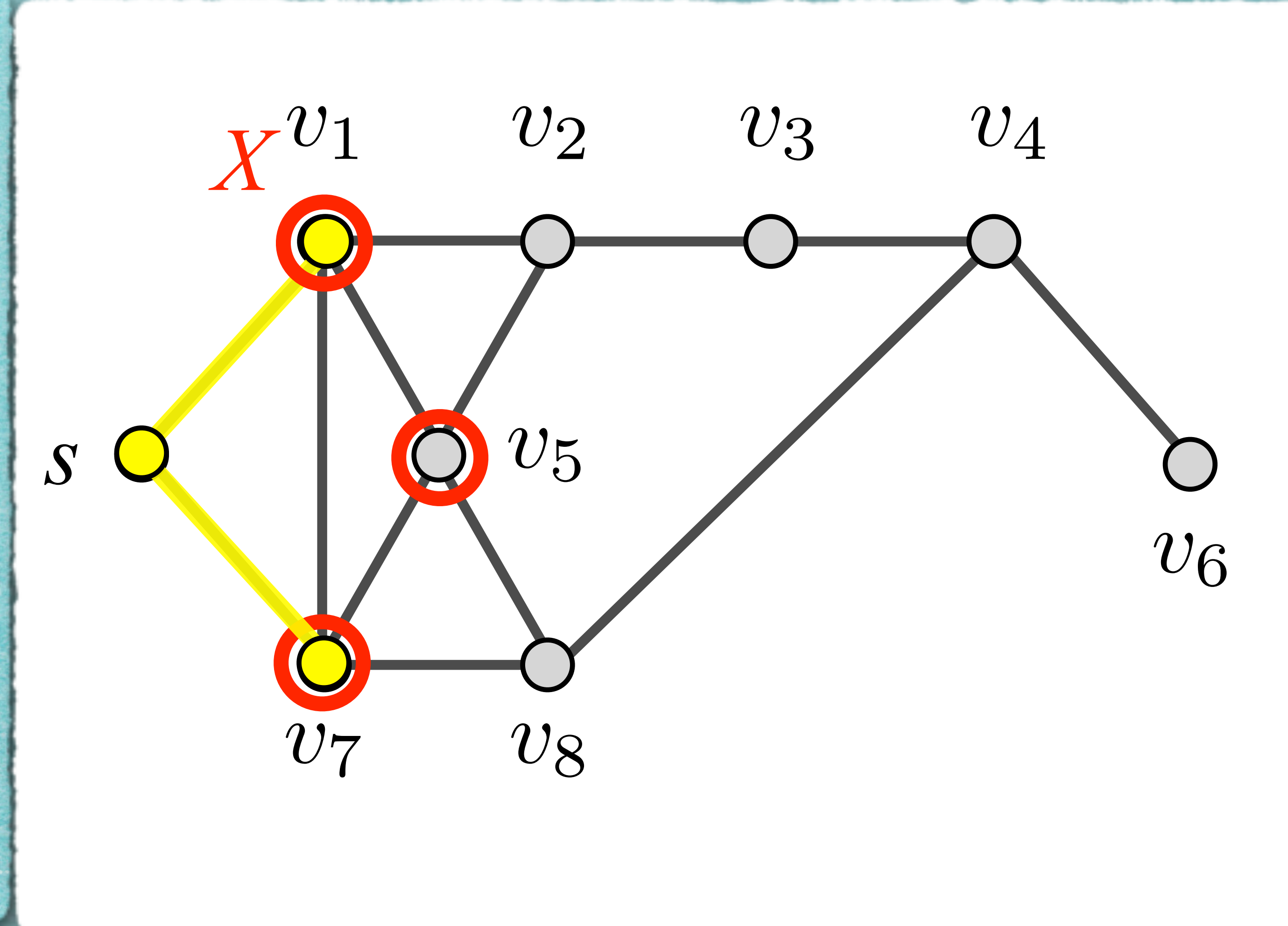
Algorithmus 3.7

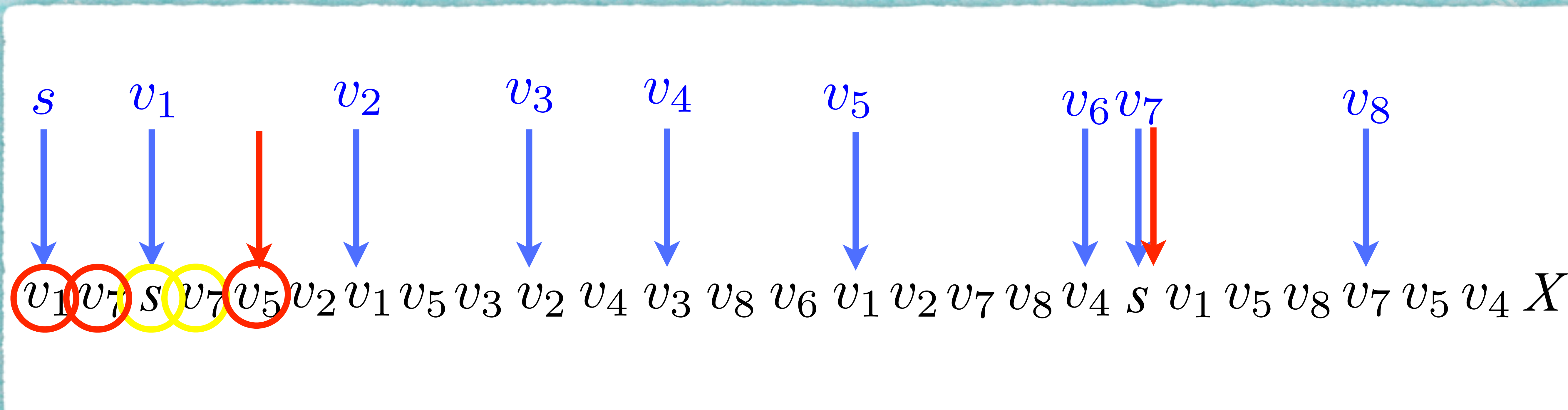
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP

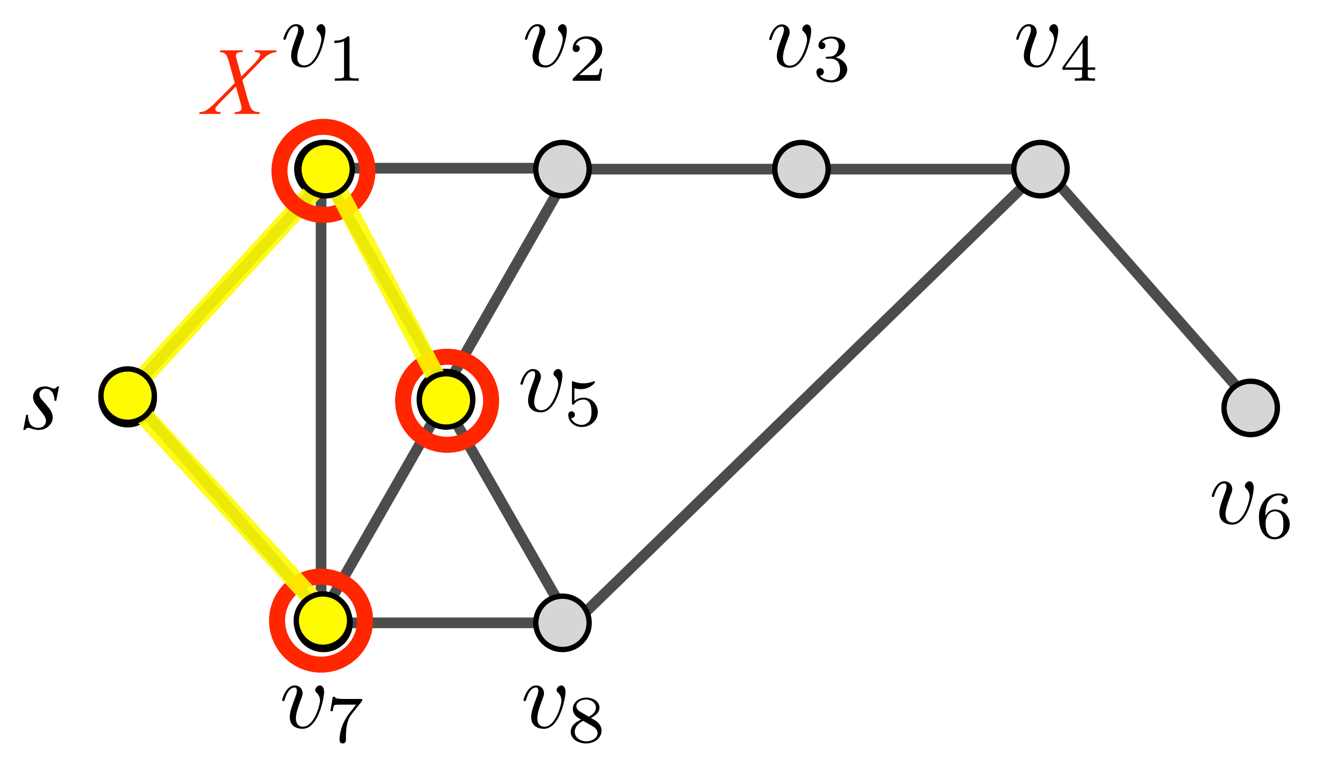


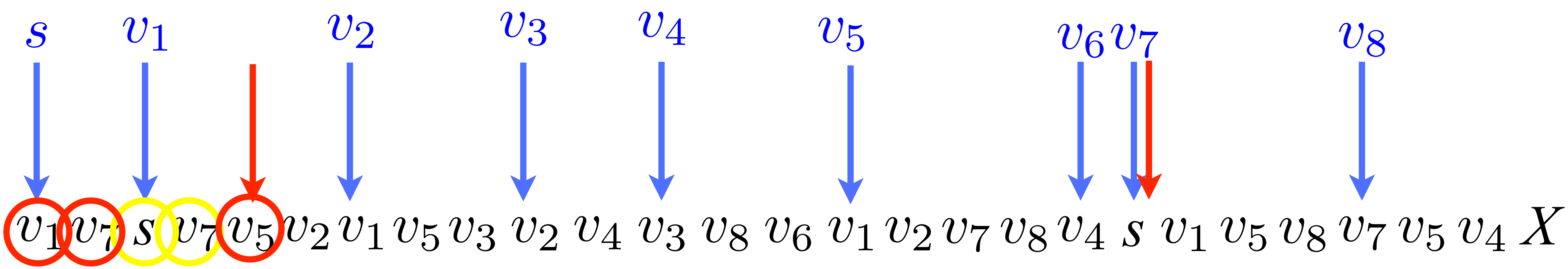


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }
    - }
3. STOP





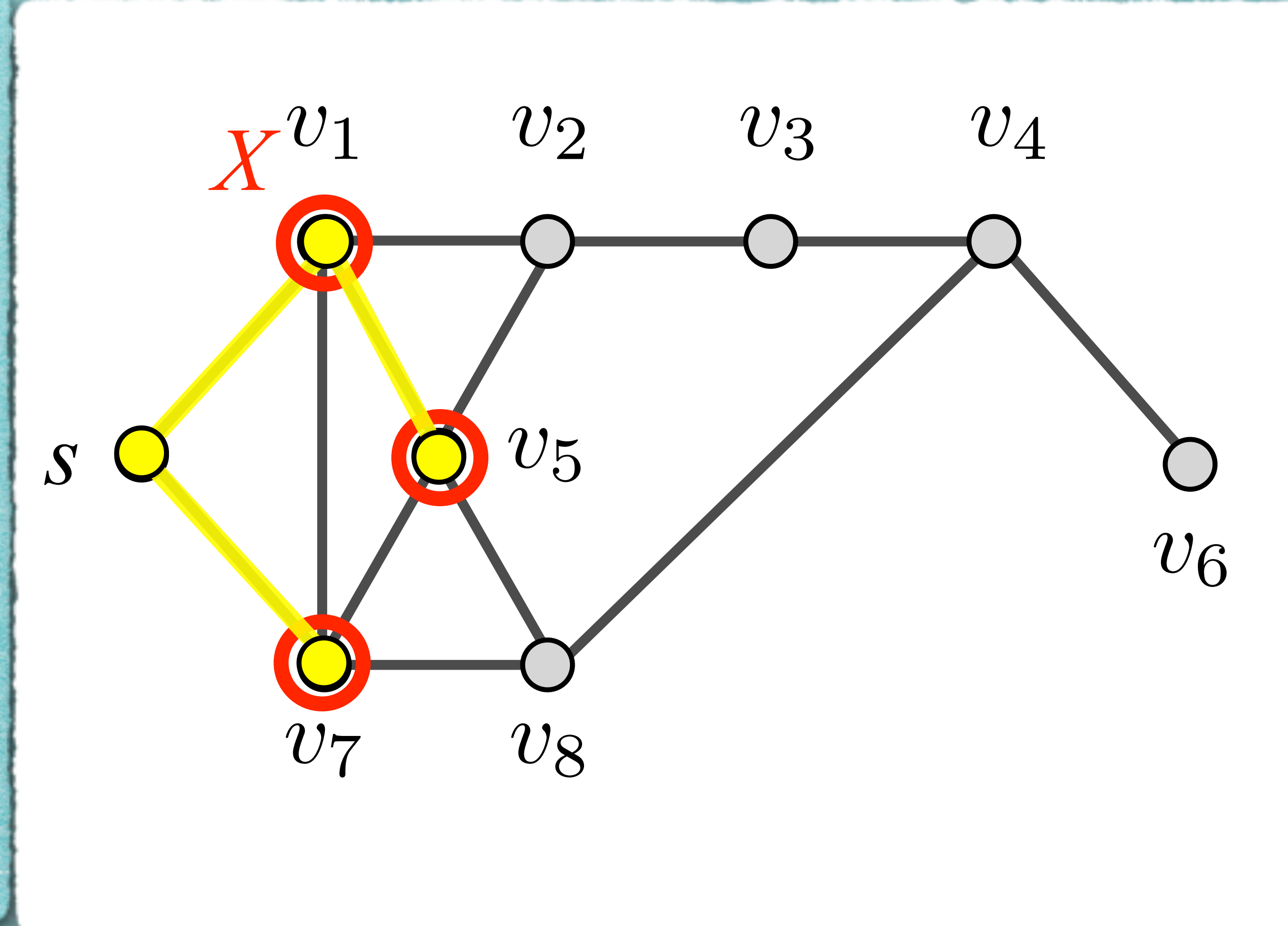
Algorithmus 3.7

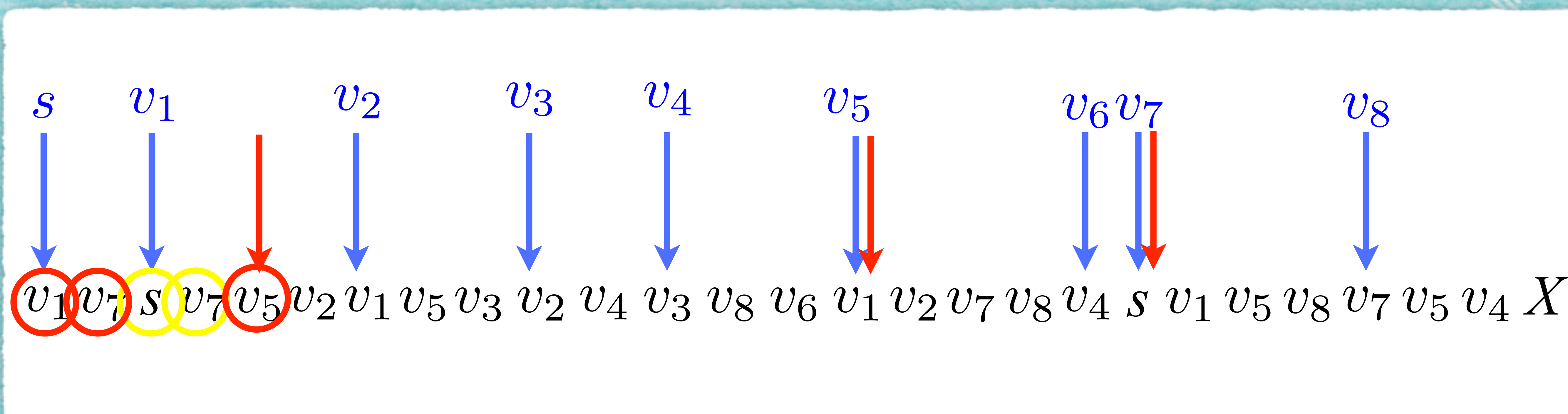
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

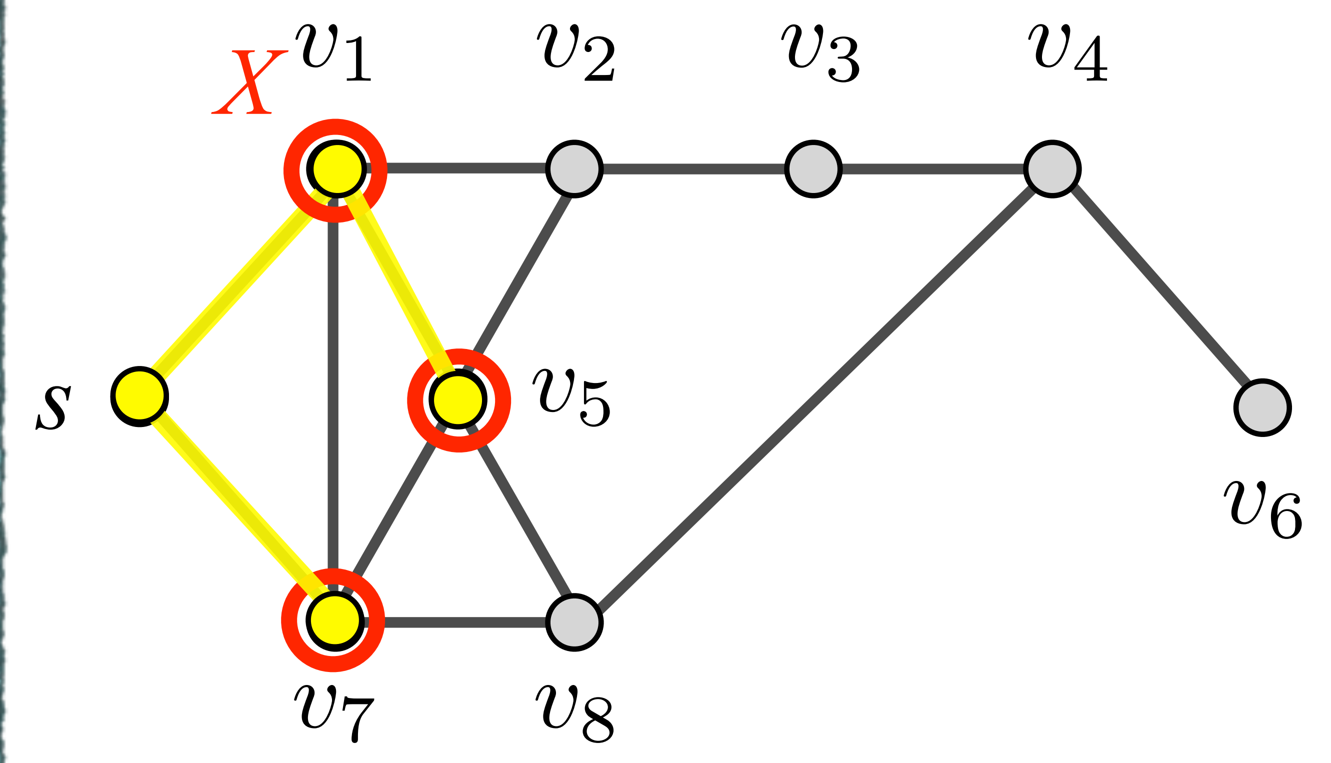
        2.3.1.Wähle ein w ∈ V\Y mit e={v,w} ∈ E

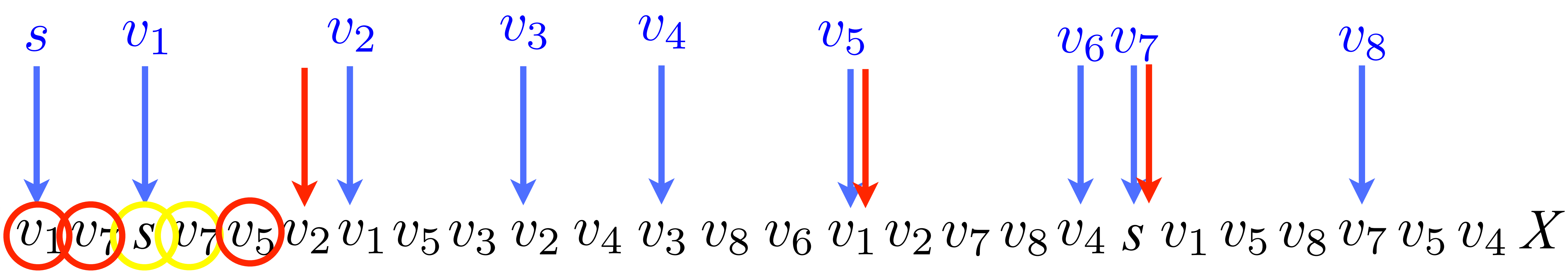
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }

3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

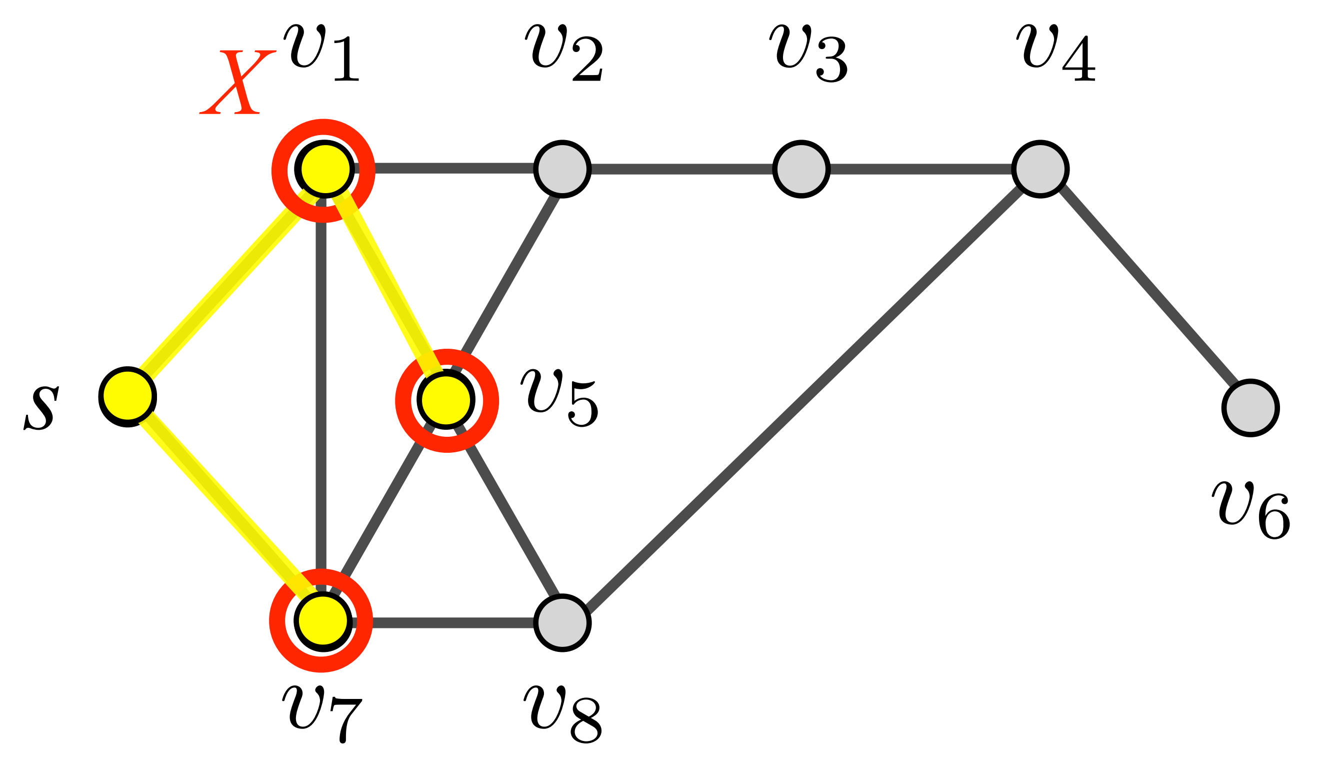
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

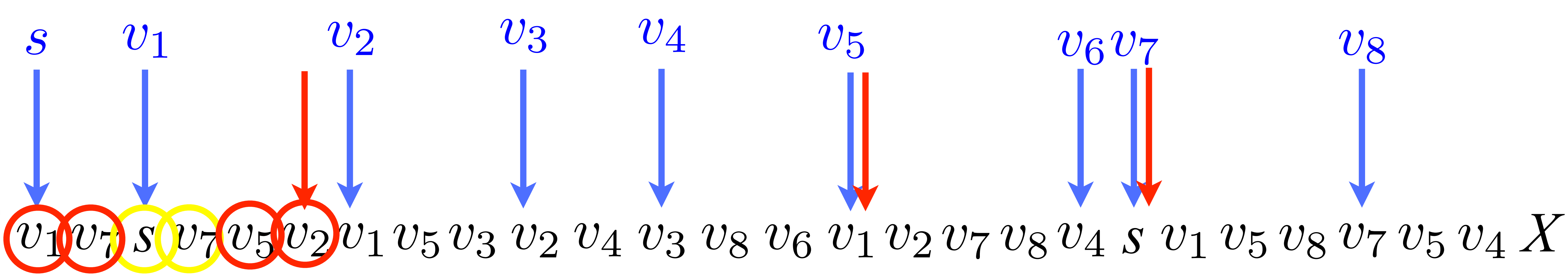
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





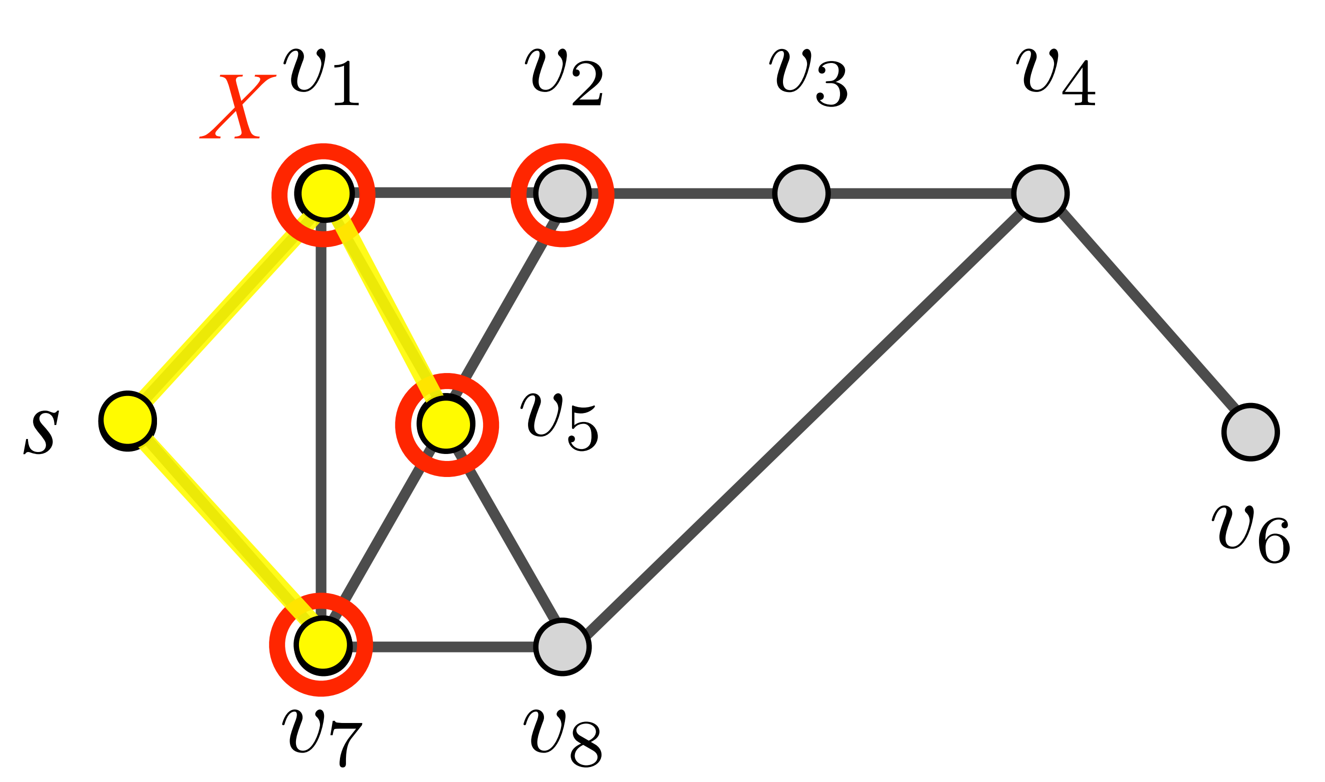
Algorithmus 3.7

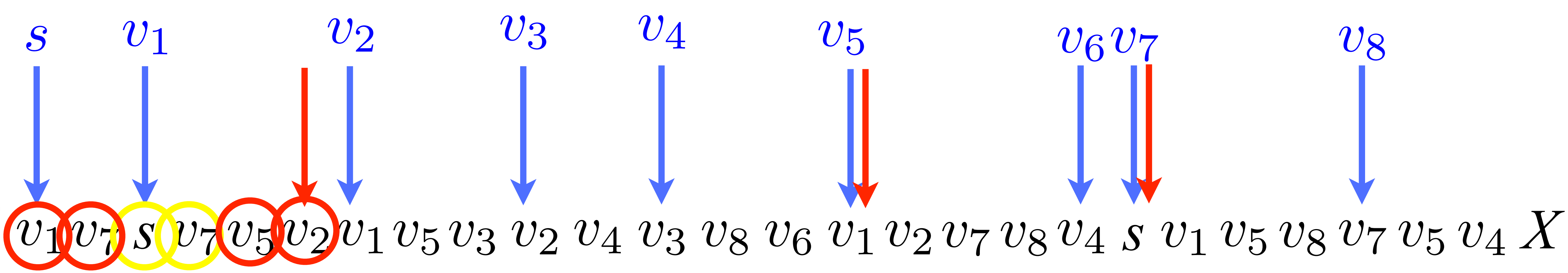
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

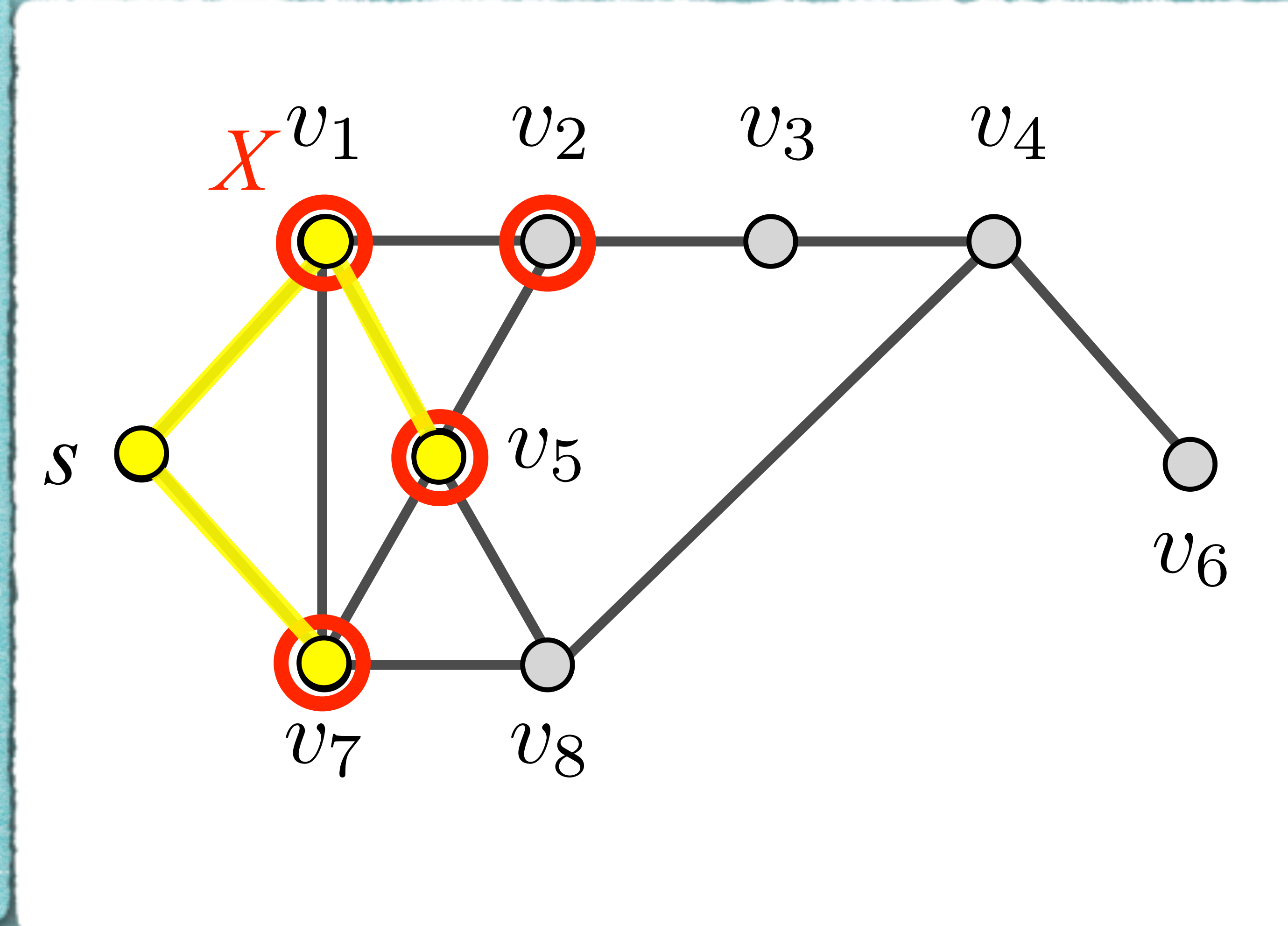
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

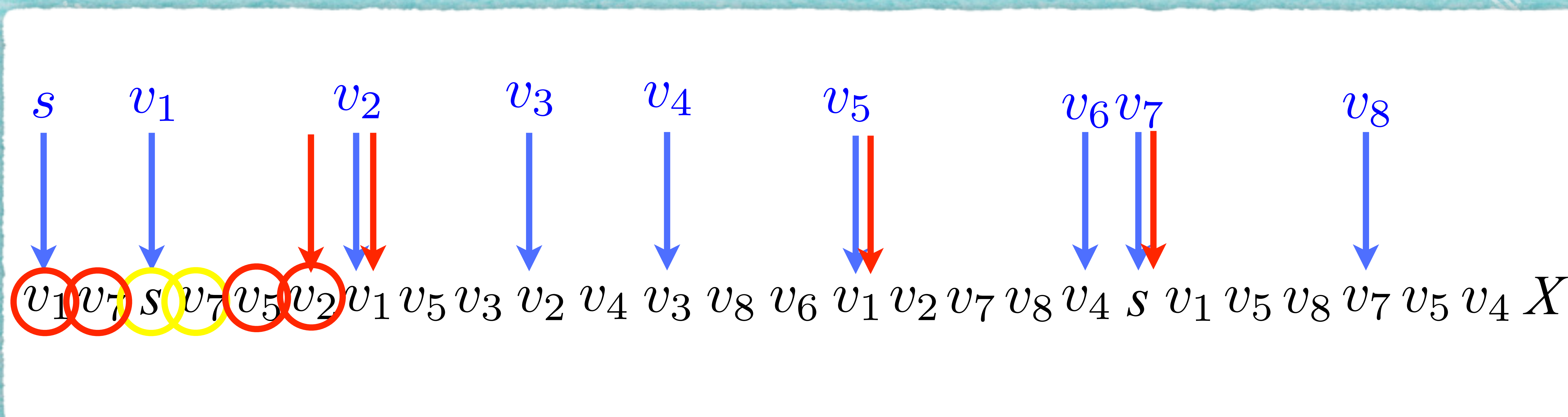
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

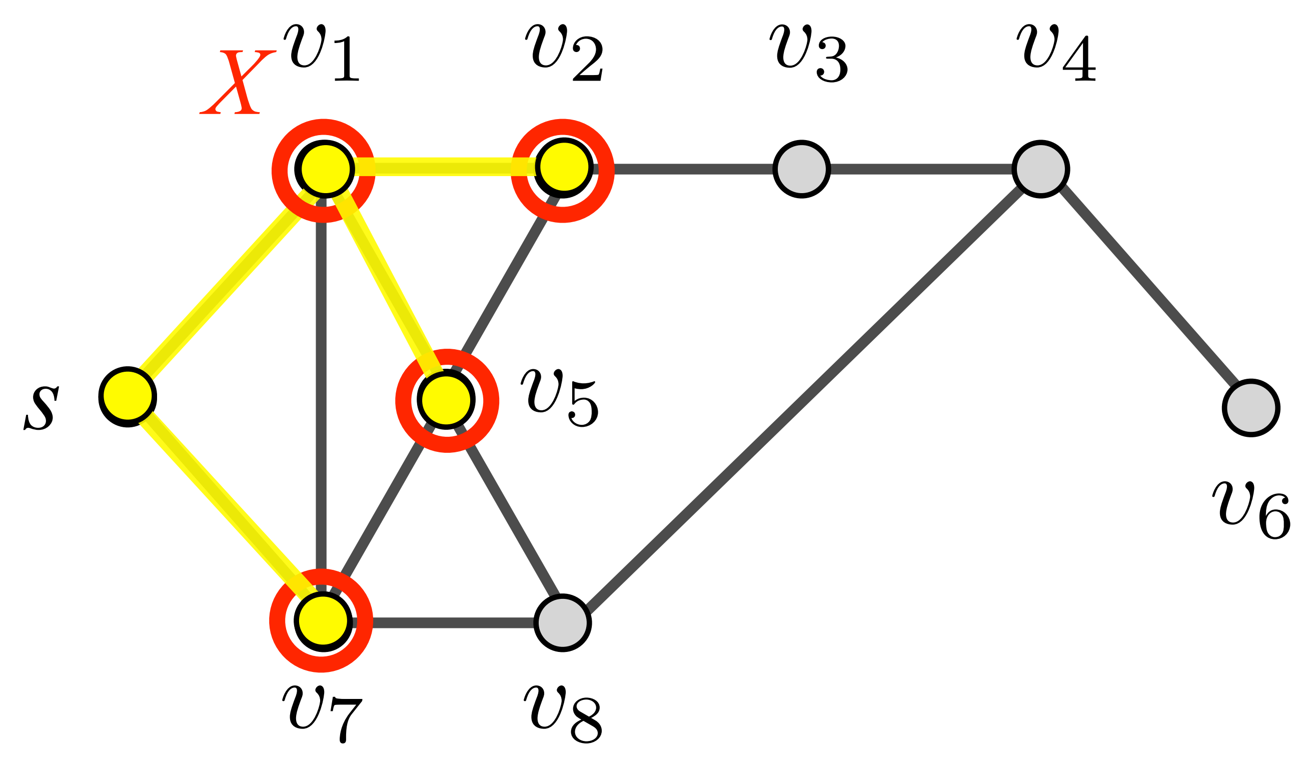
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

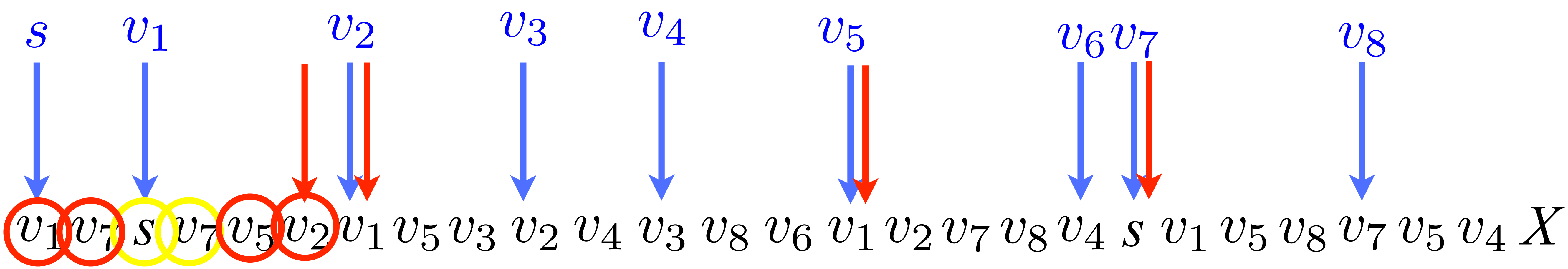
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE $(R \neq \emptyset)$ DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





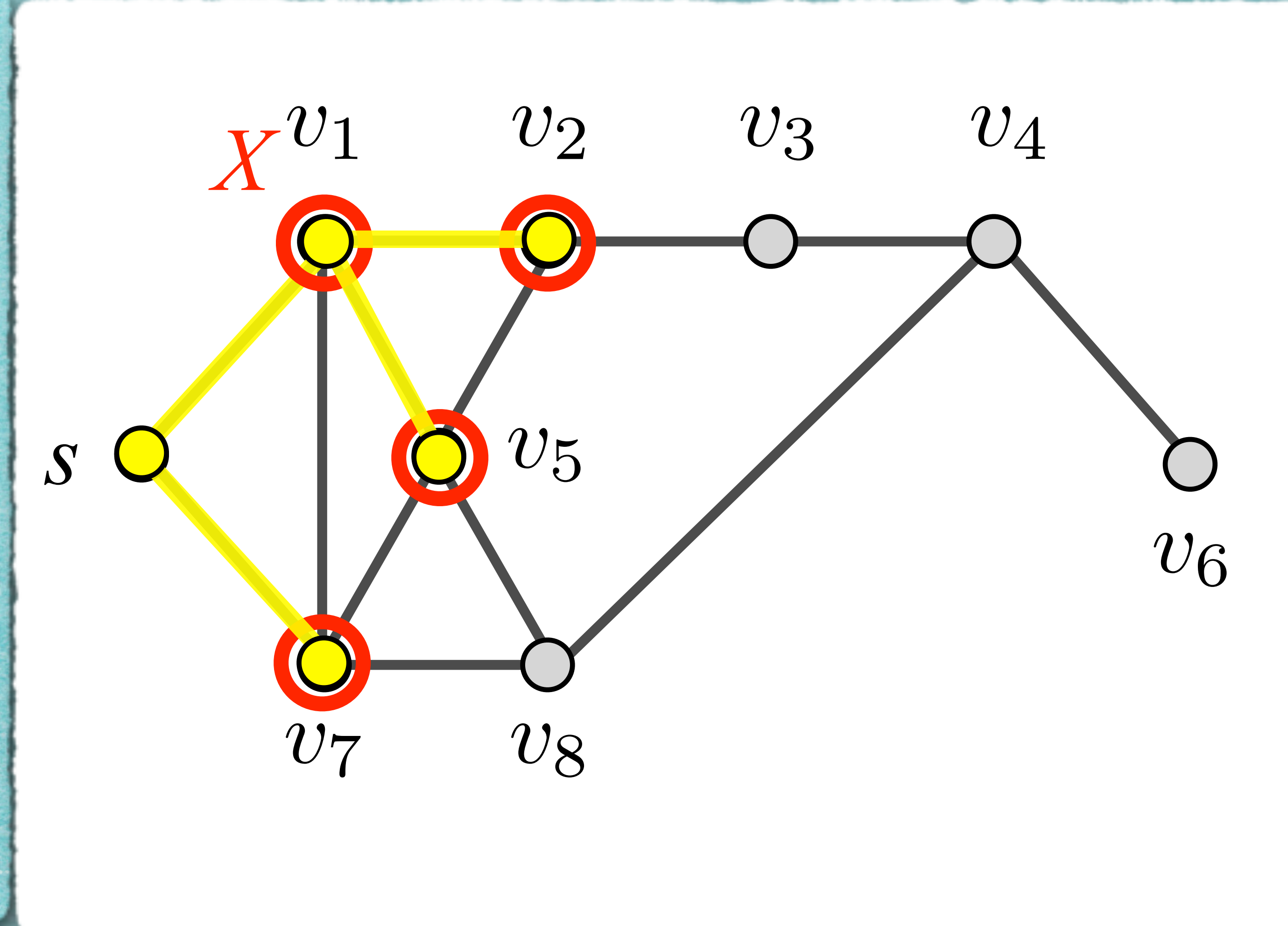
Algorithmus 3.7

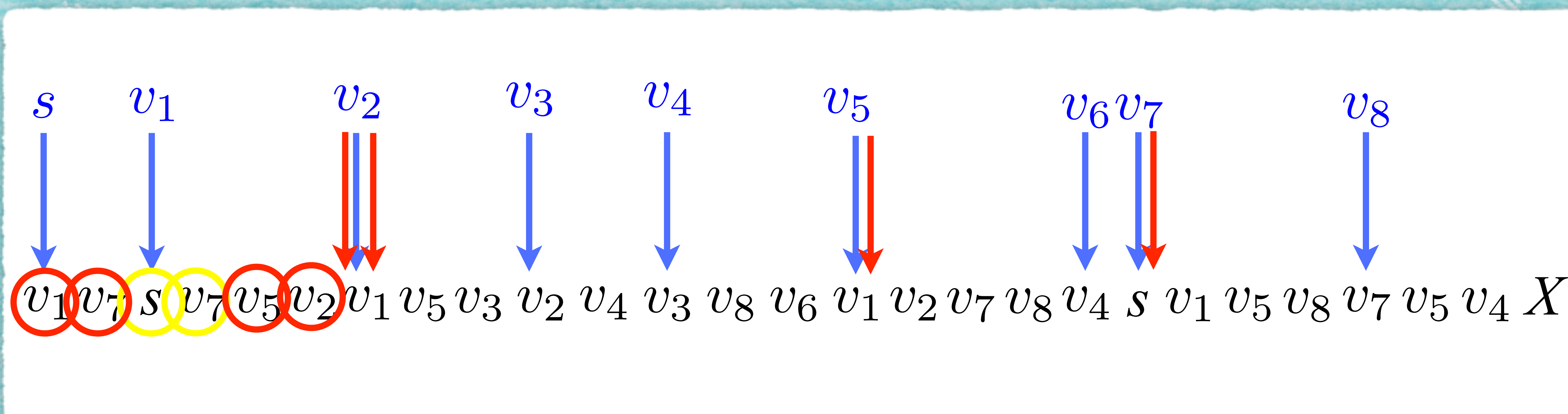
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }
    - }
3. STOP





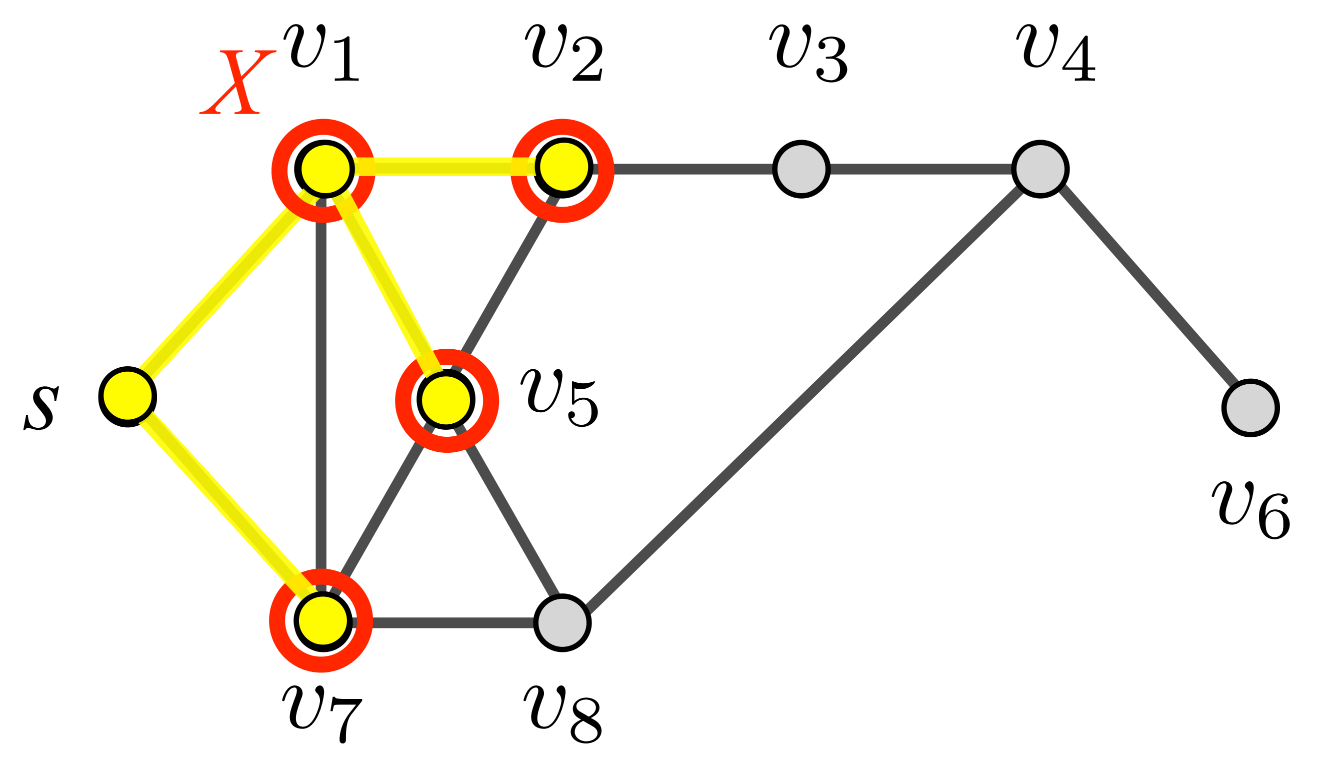
Algorithmus 3.7

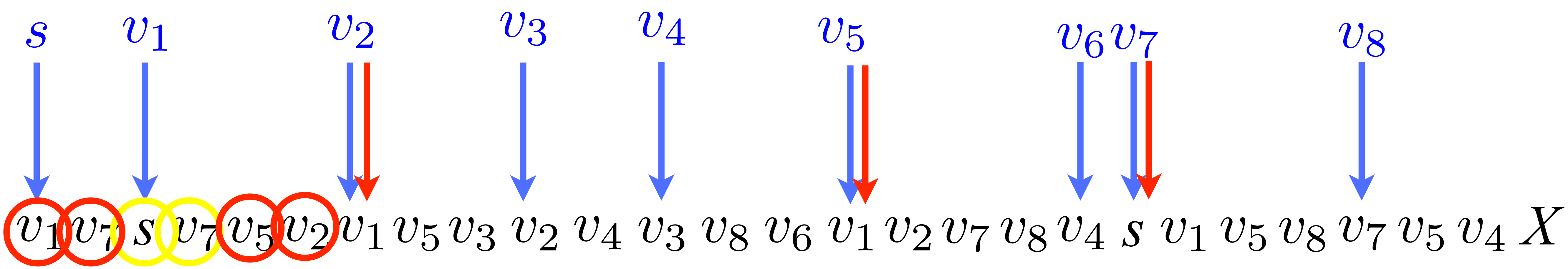
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

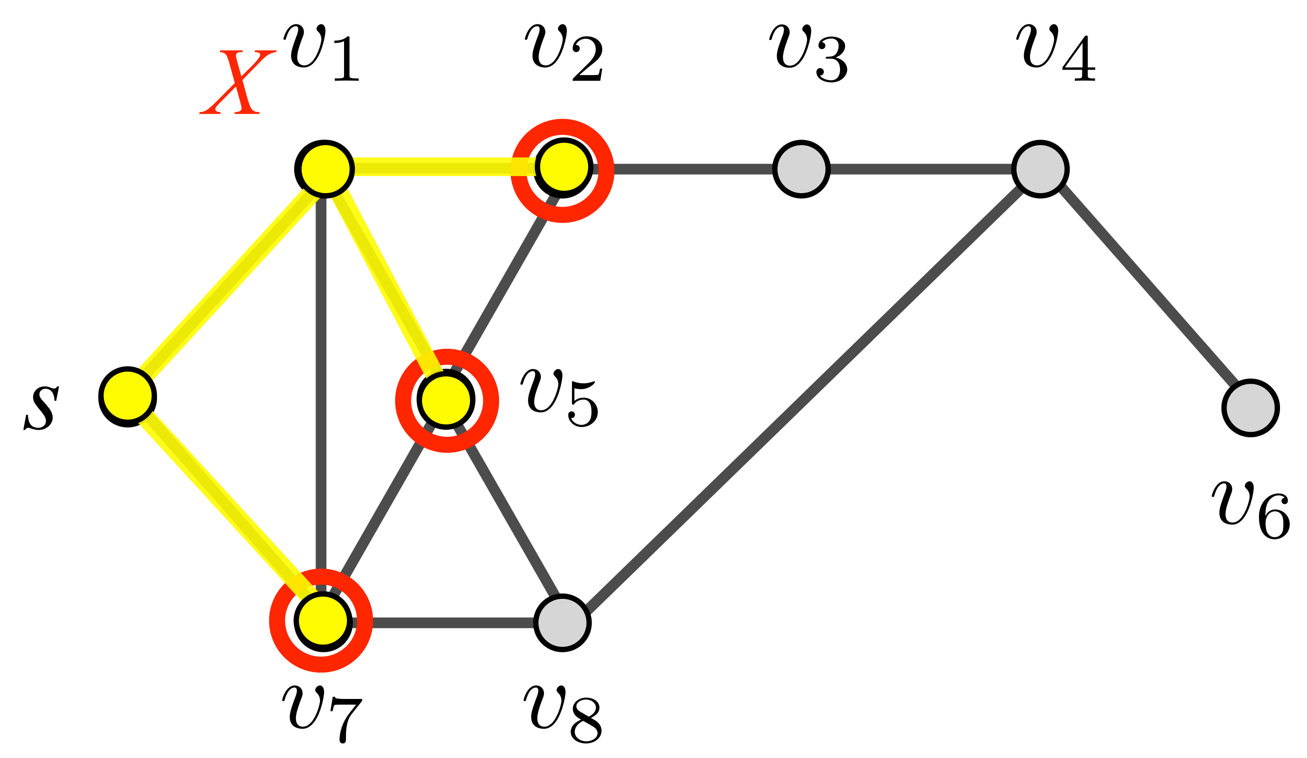
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

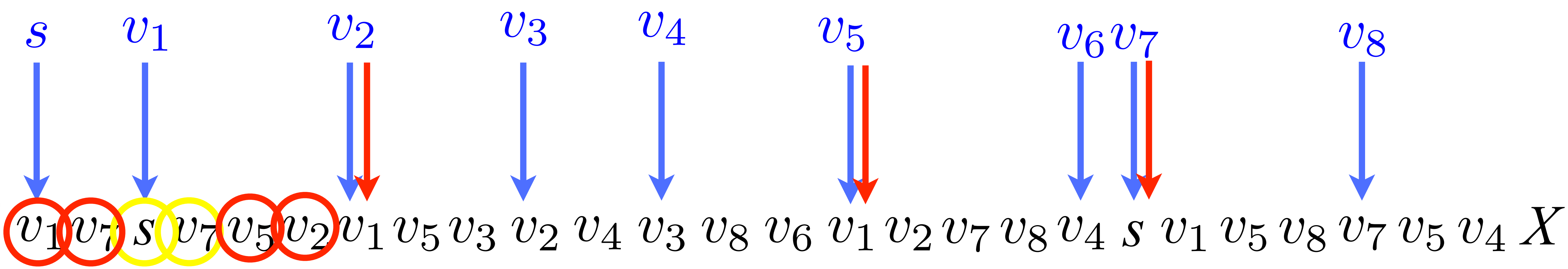
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
        - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

        2.2.1. R:=R\{v}

    2.3. ELSE {

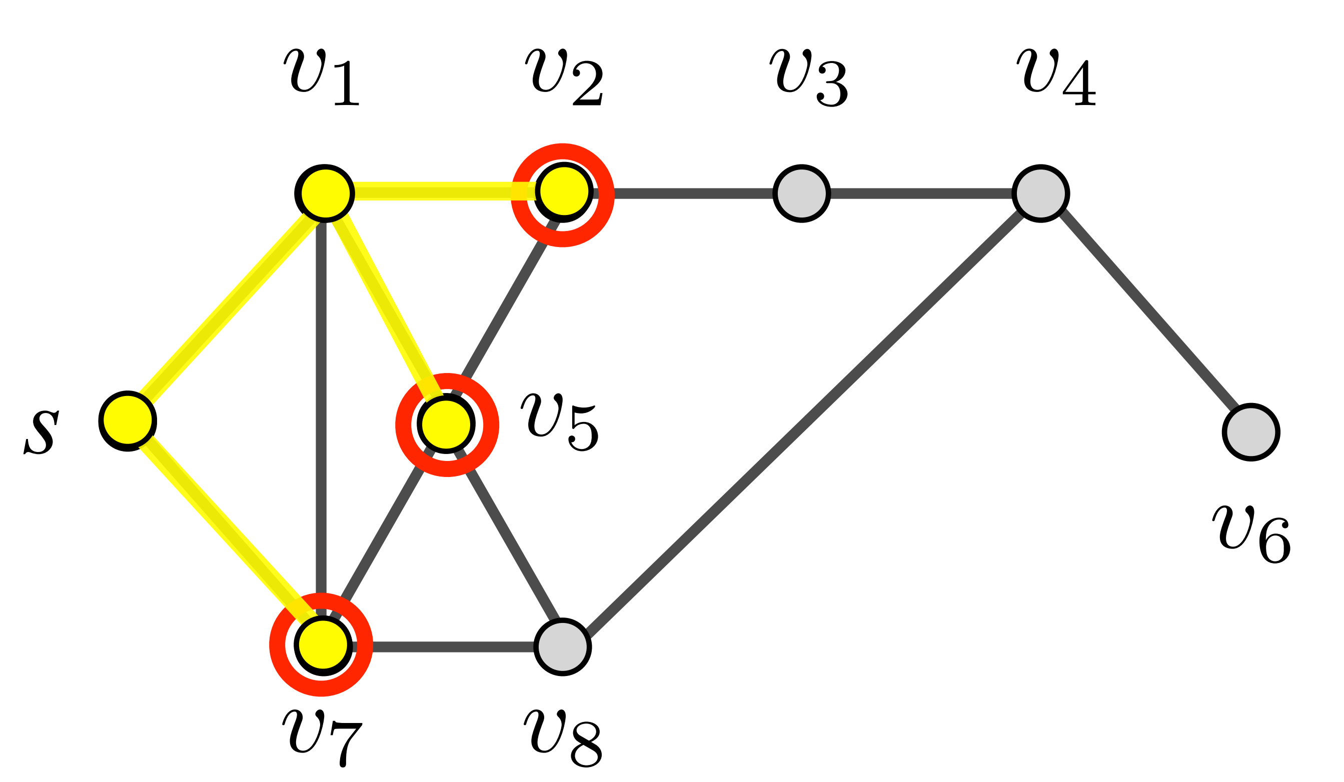
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

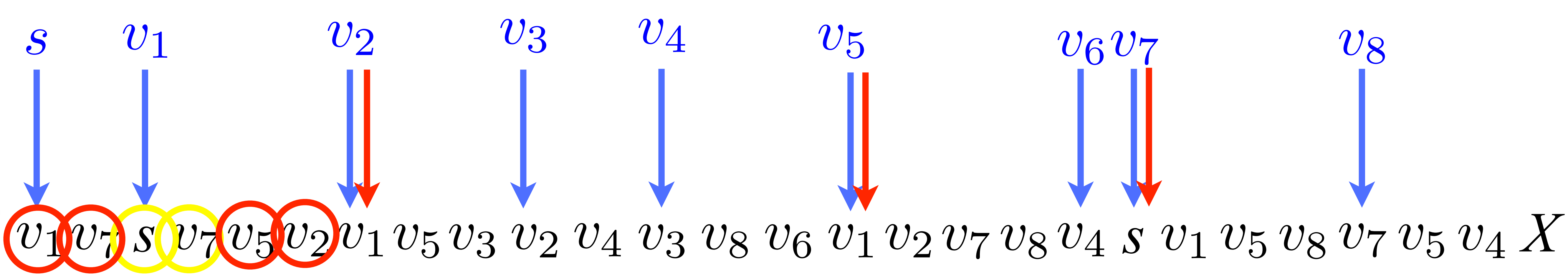
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }

3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

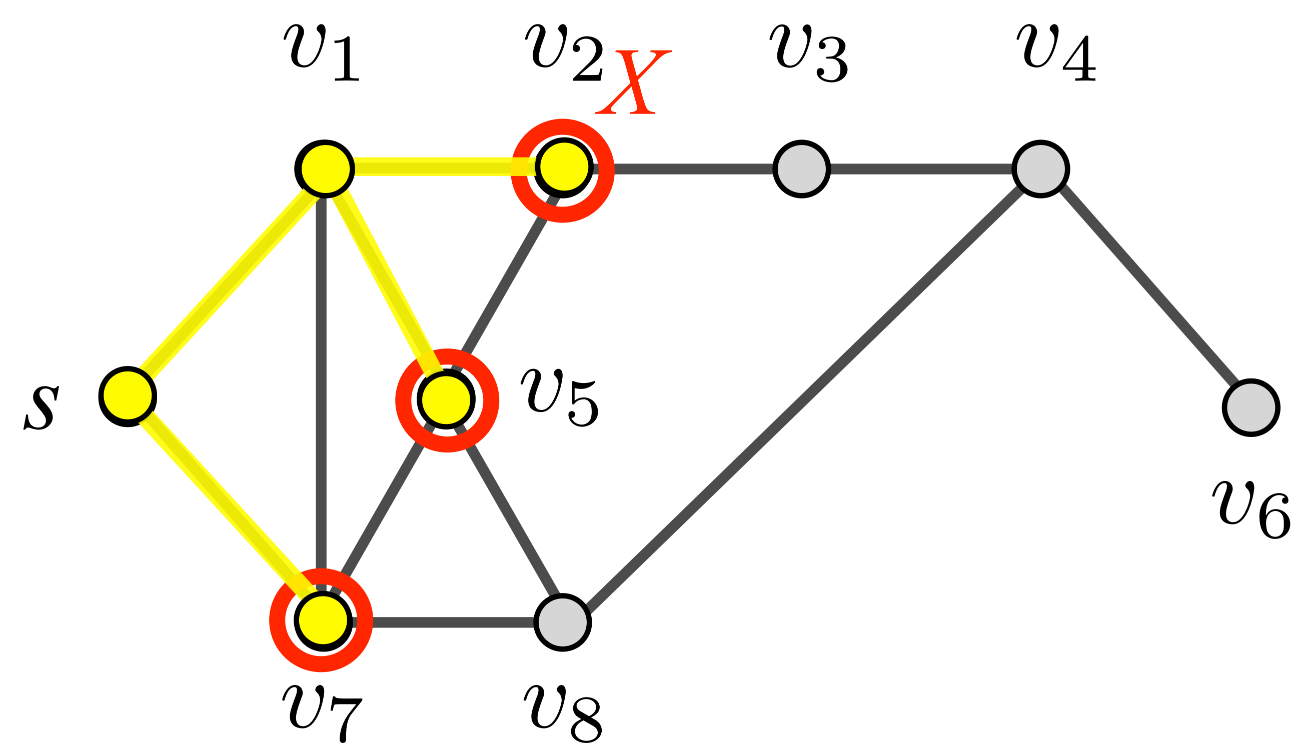
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

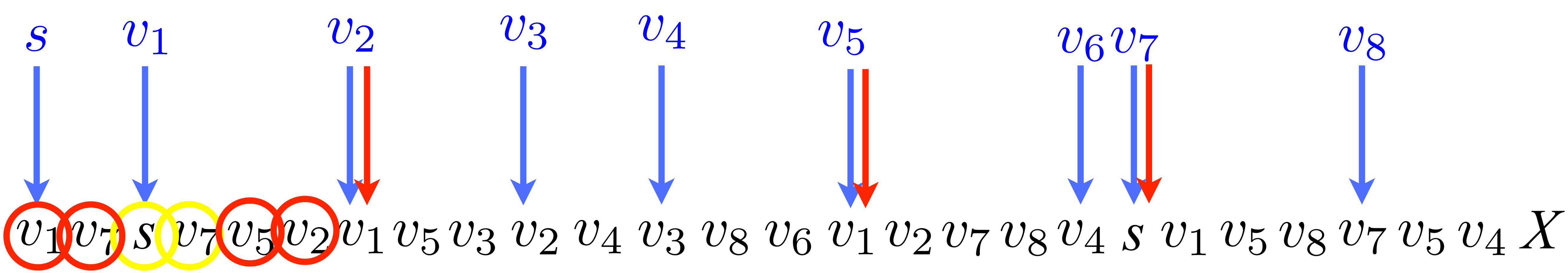
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

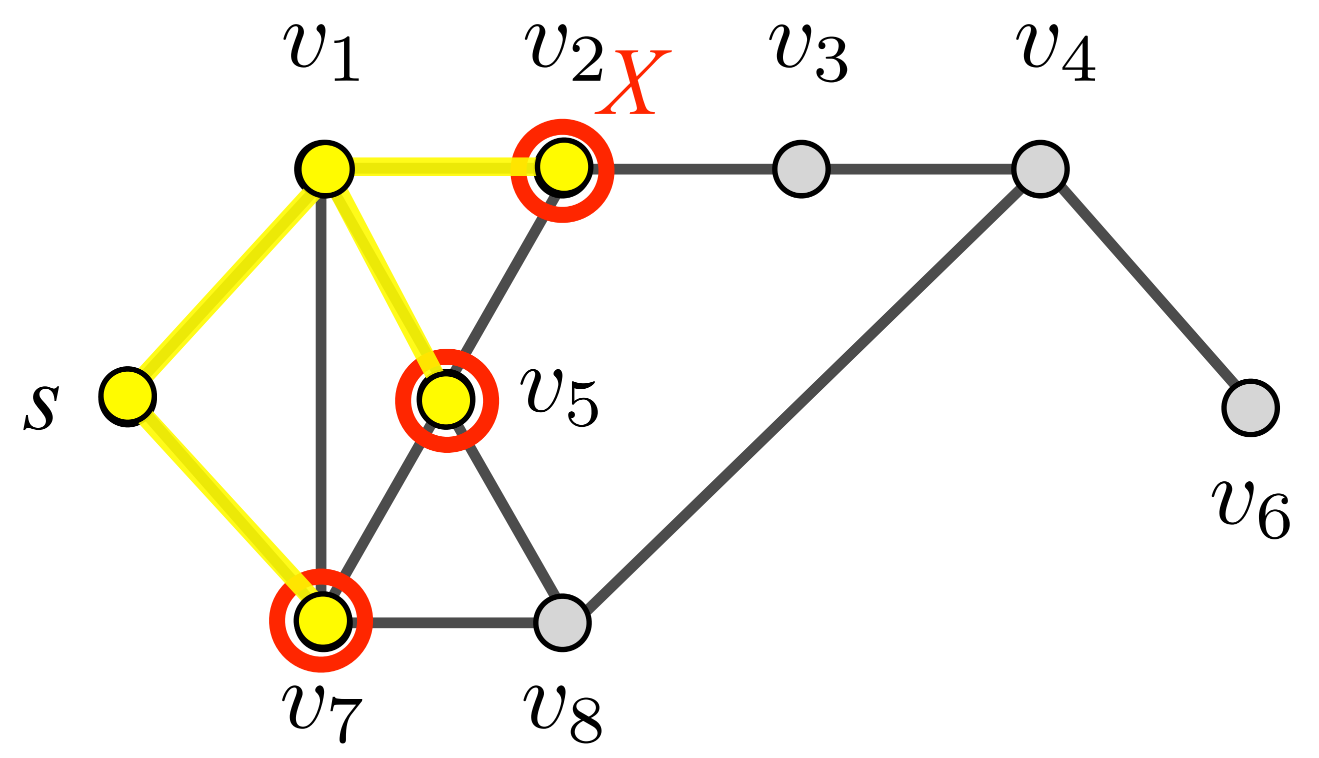


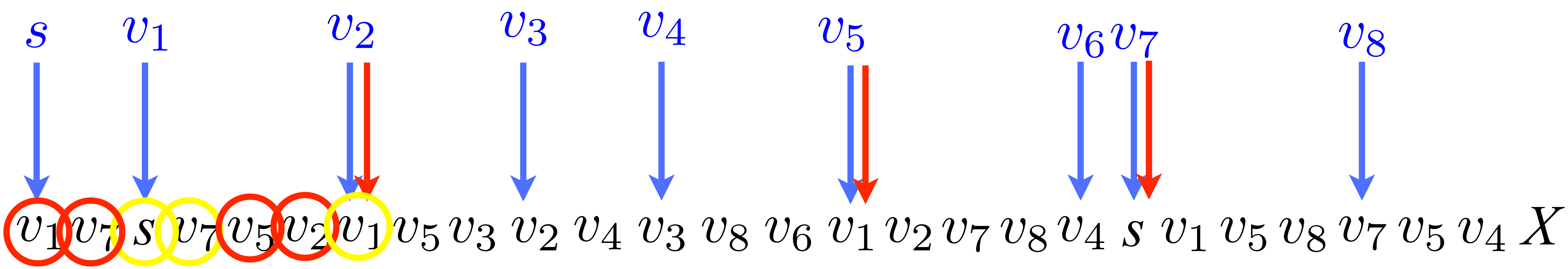


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
   }
3. STOP

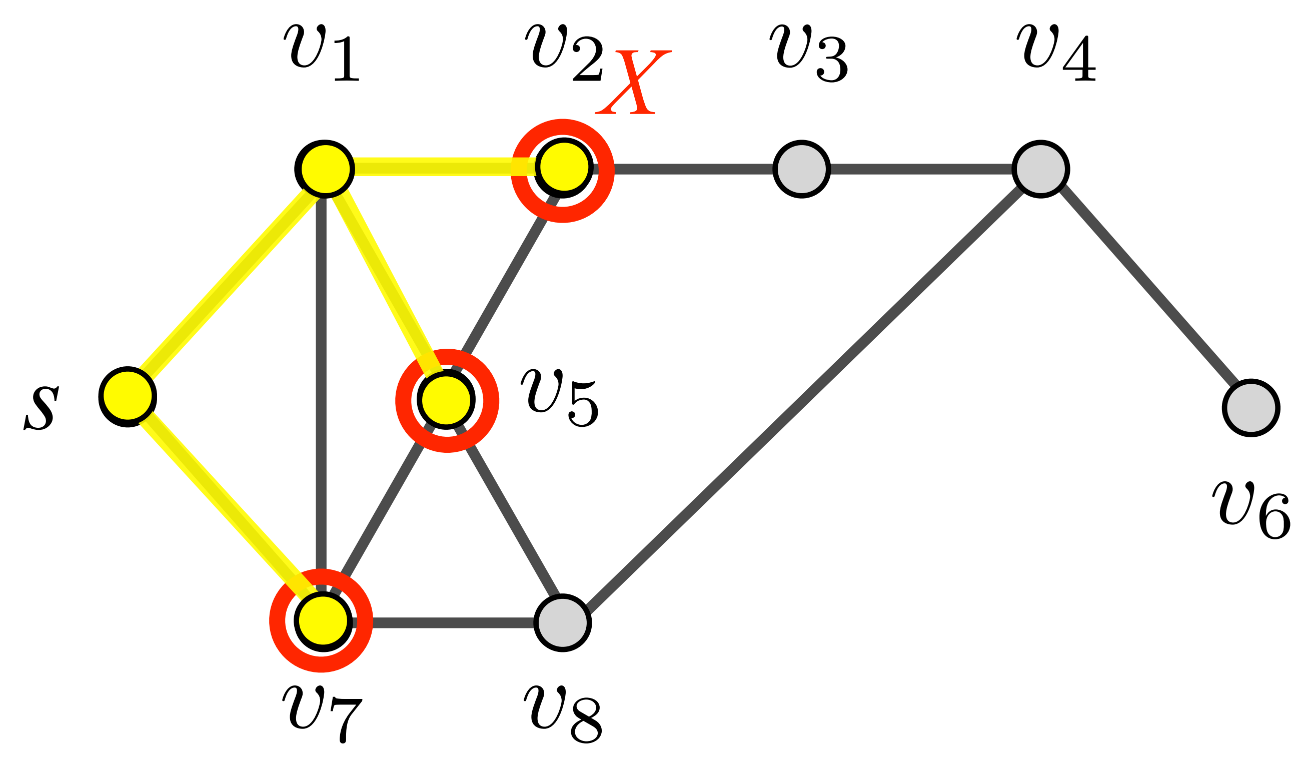


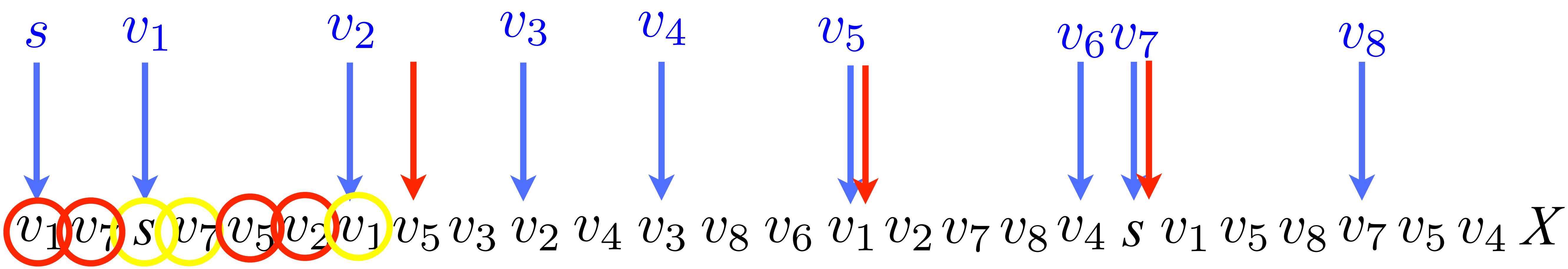


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

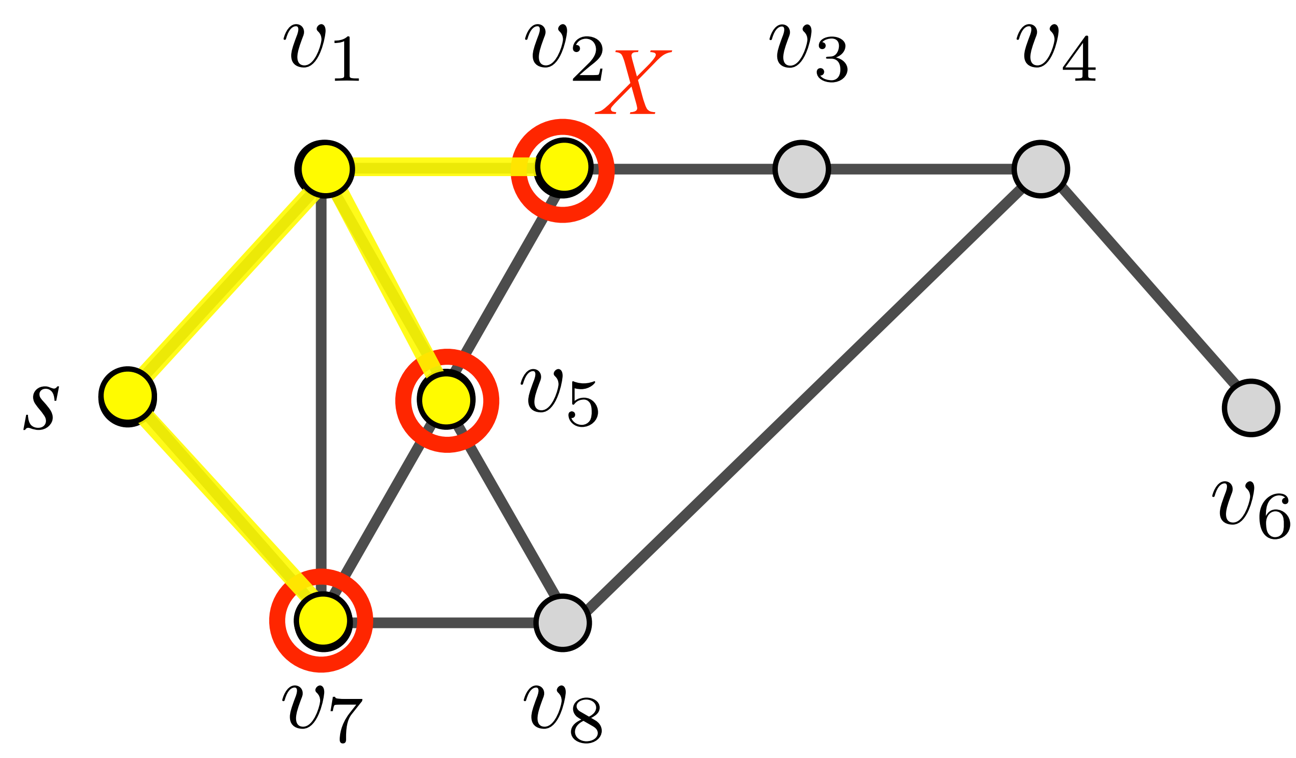
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

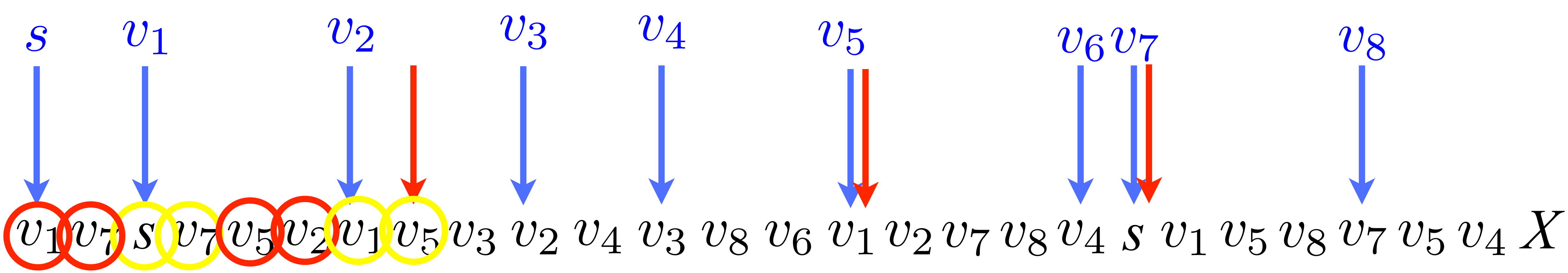
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP





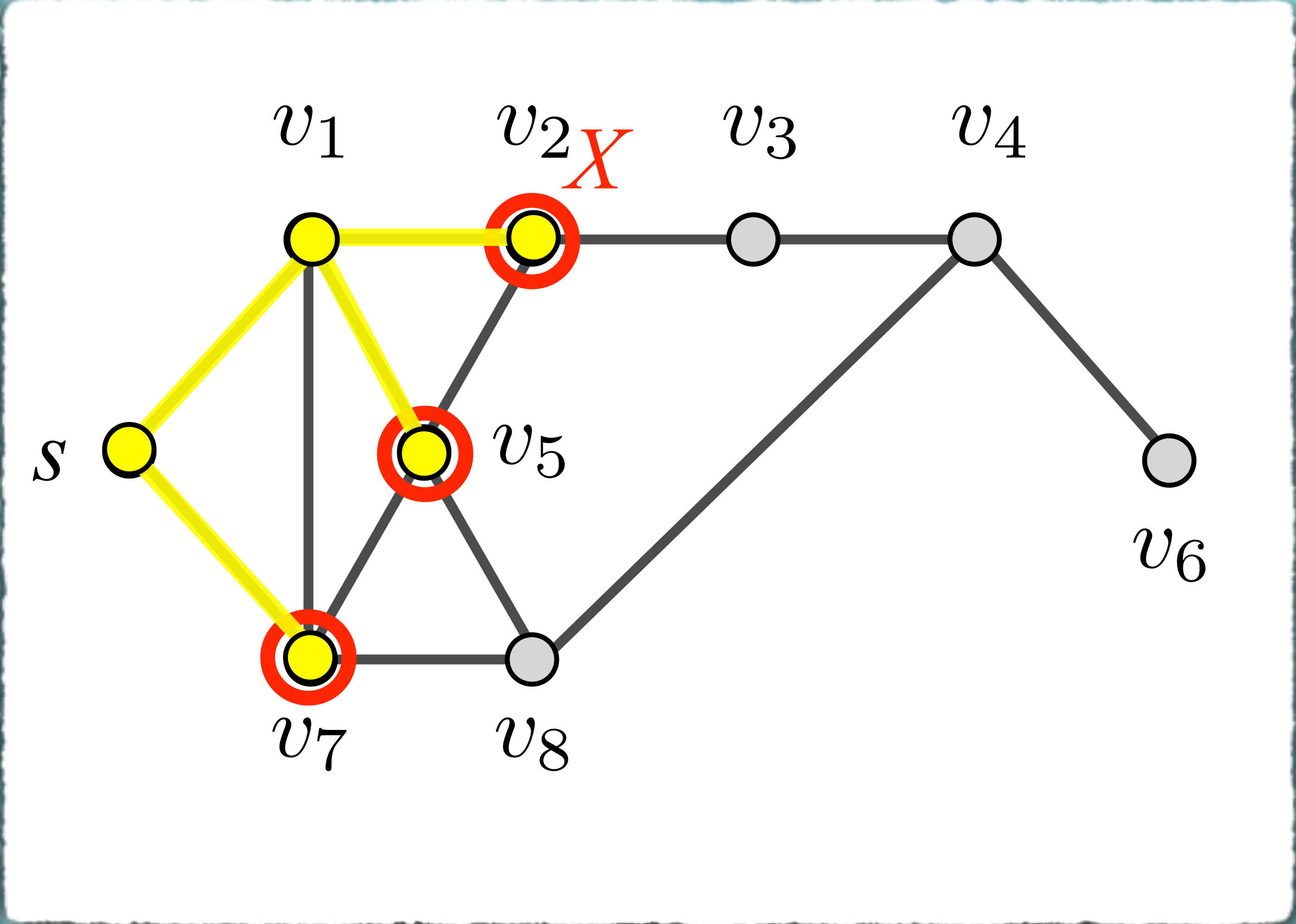
Algorithmus 3.7

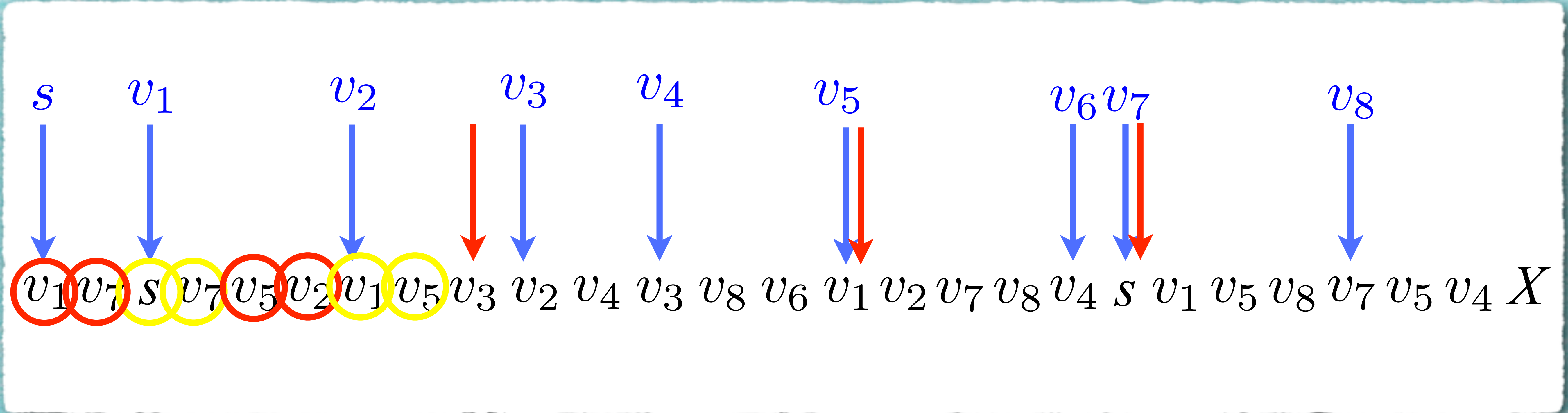
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP





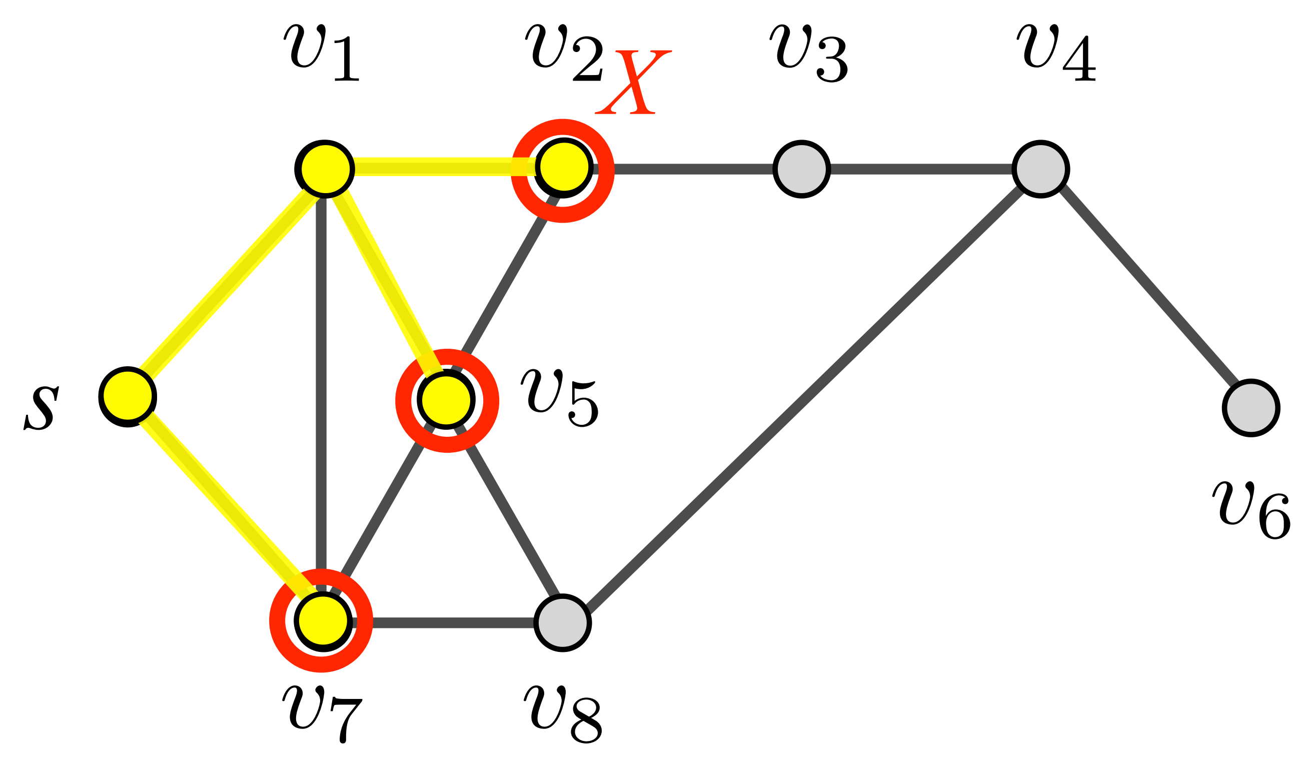
Algorithmus 3.7

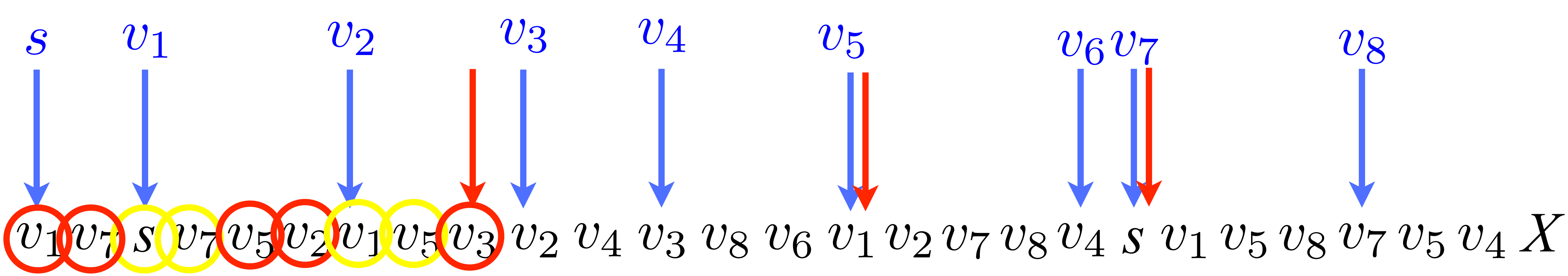
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

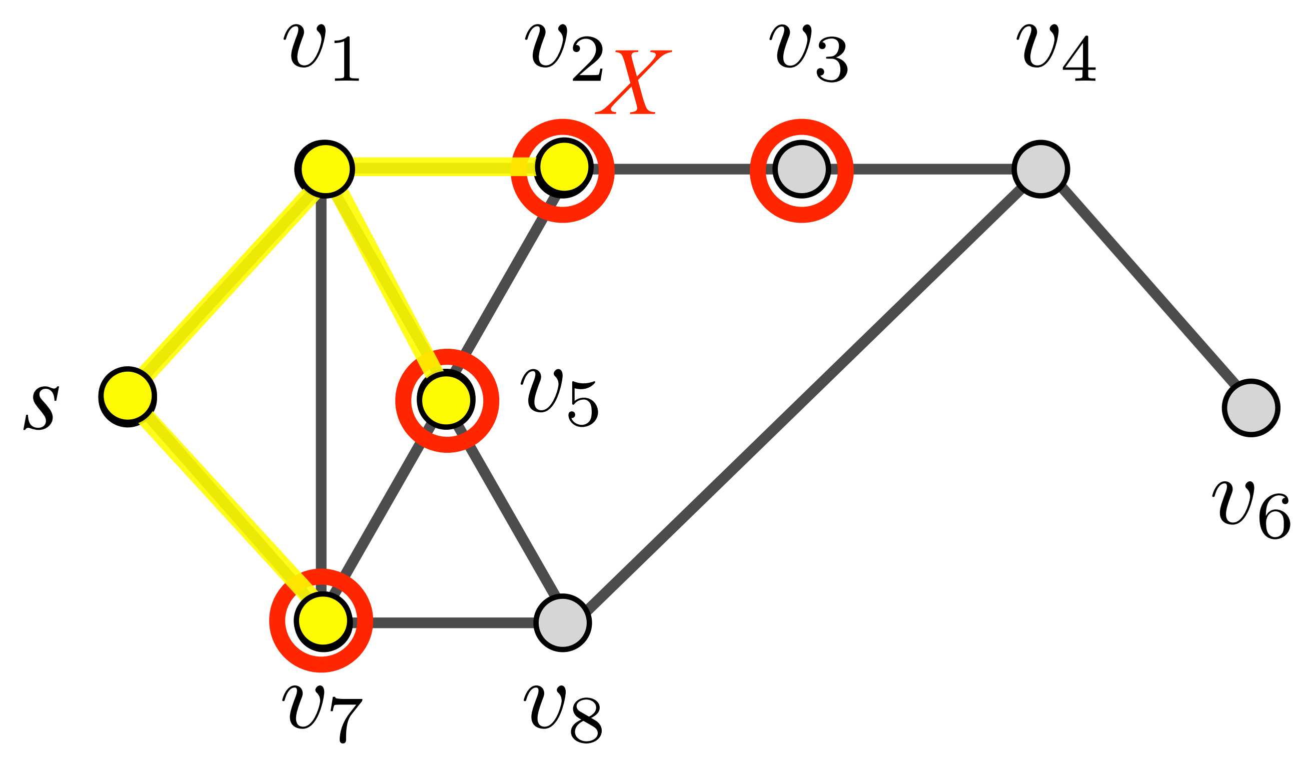
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

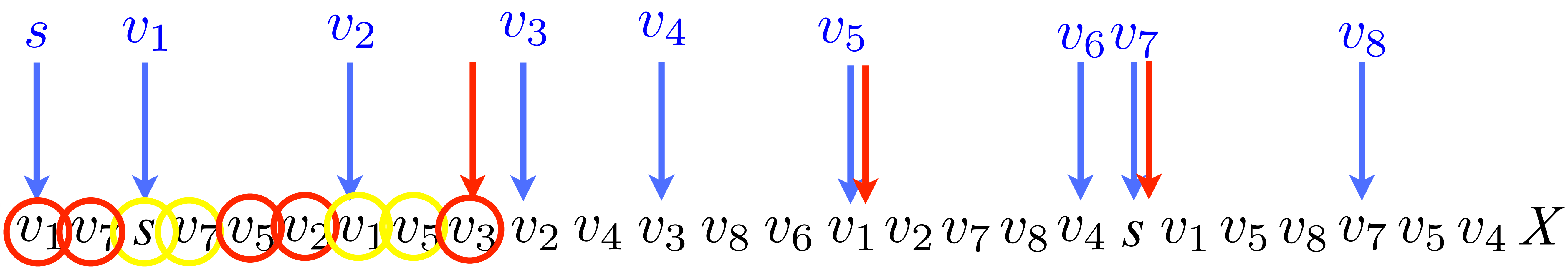
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

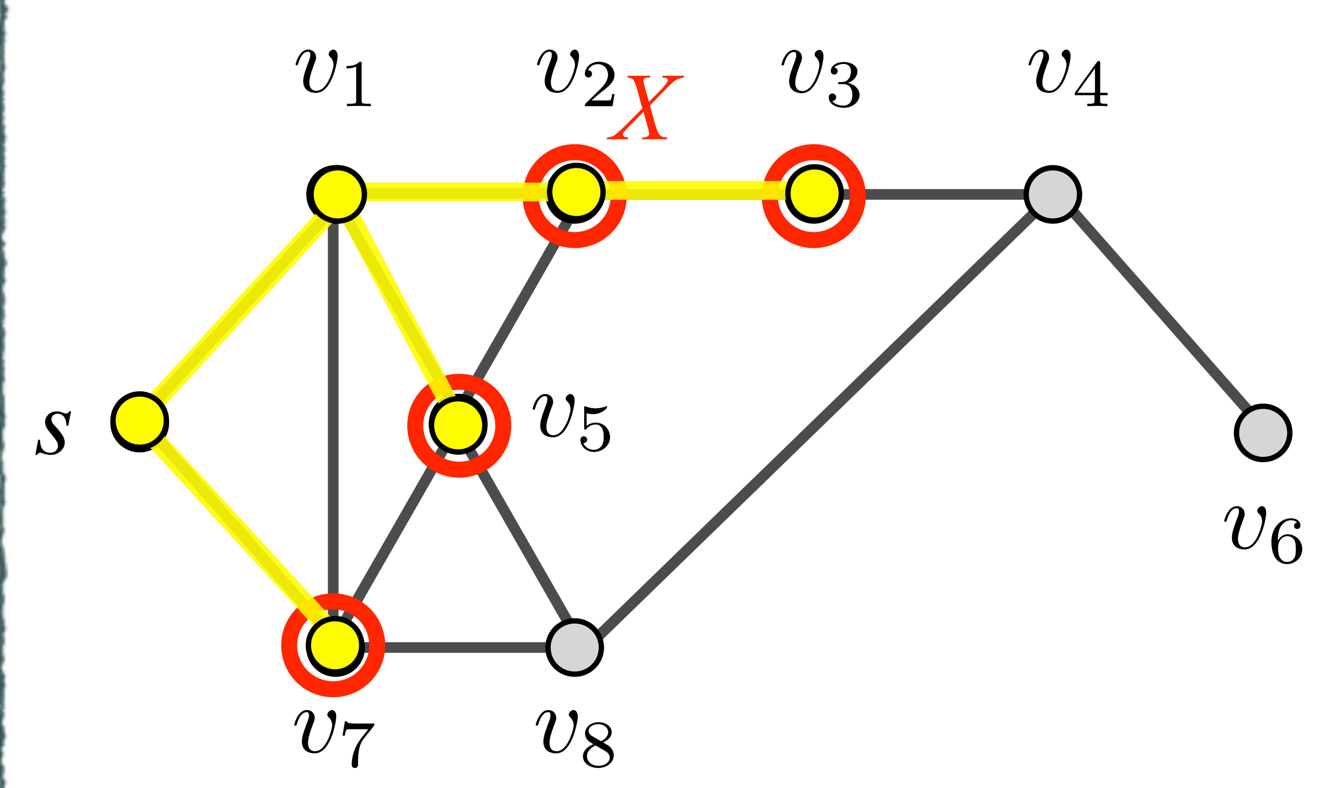


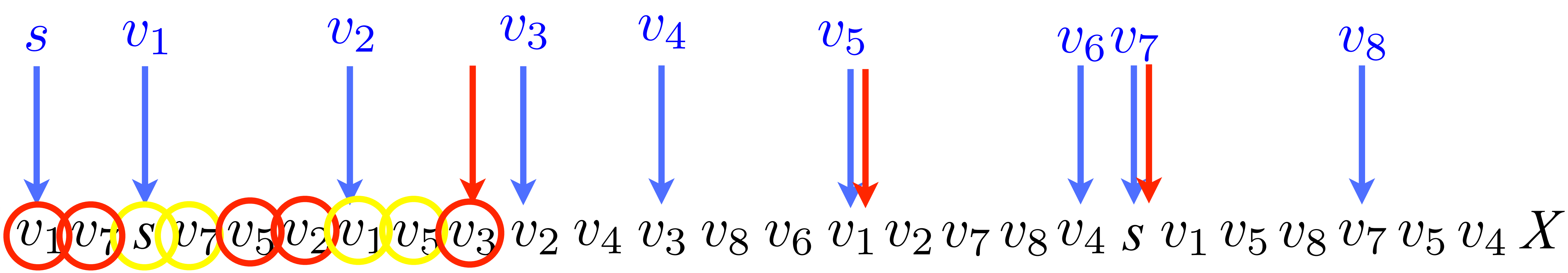


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

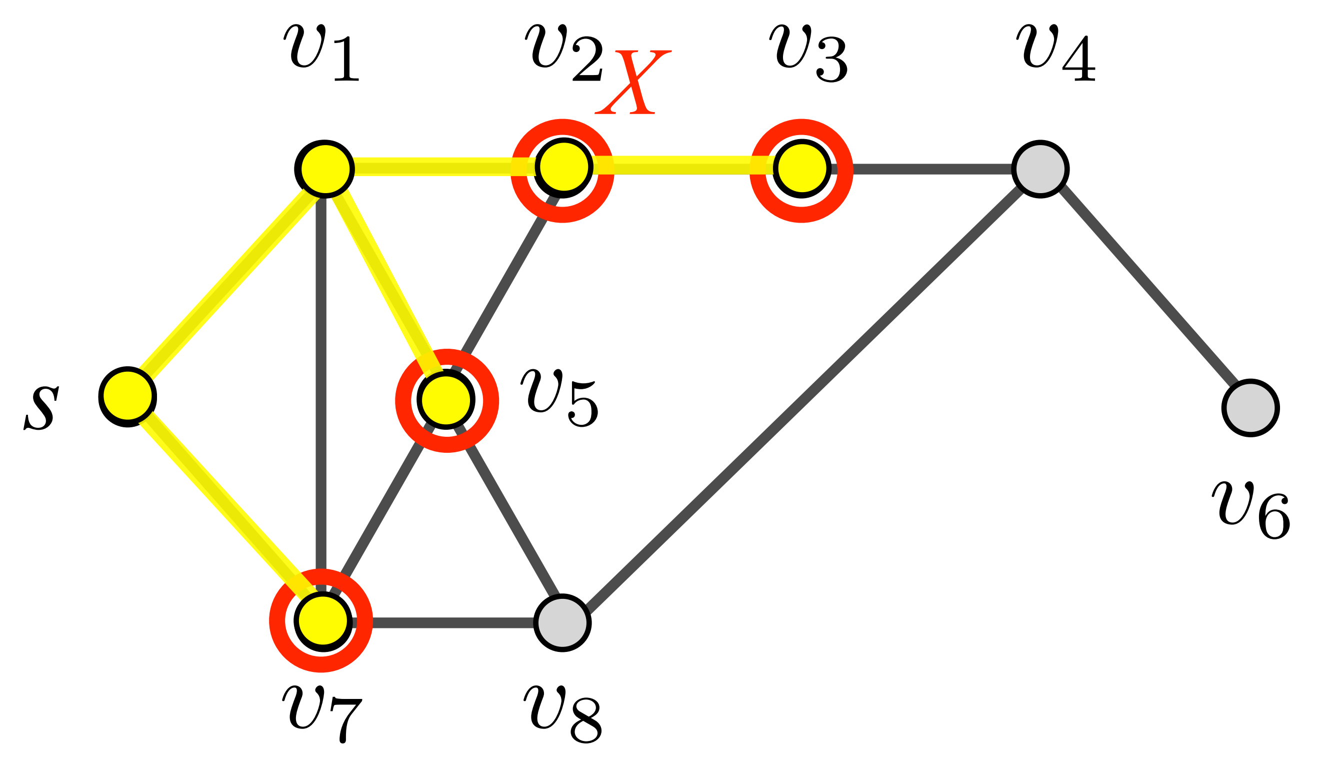
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

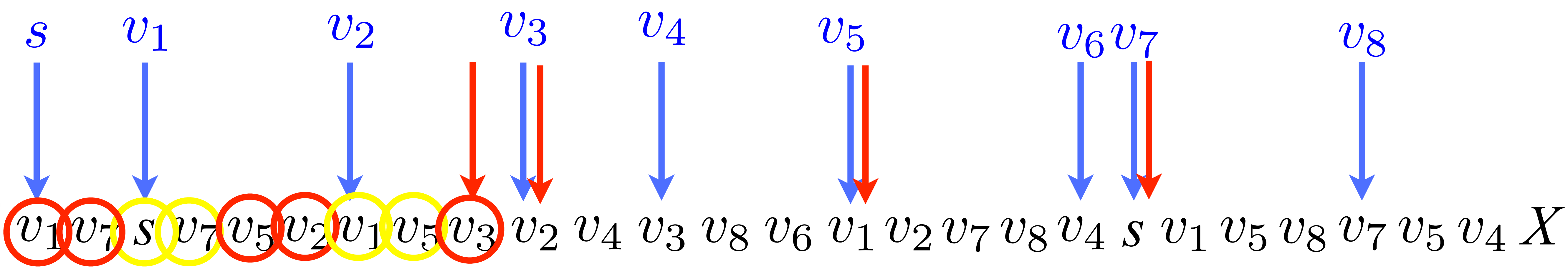
1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R\setminus\{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP

2





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

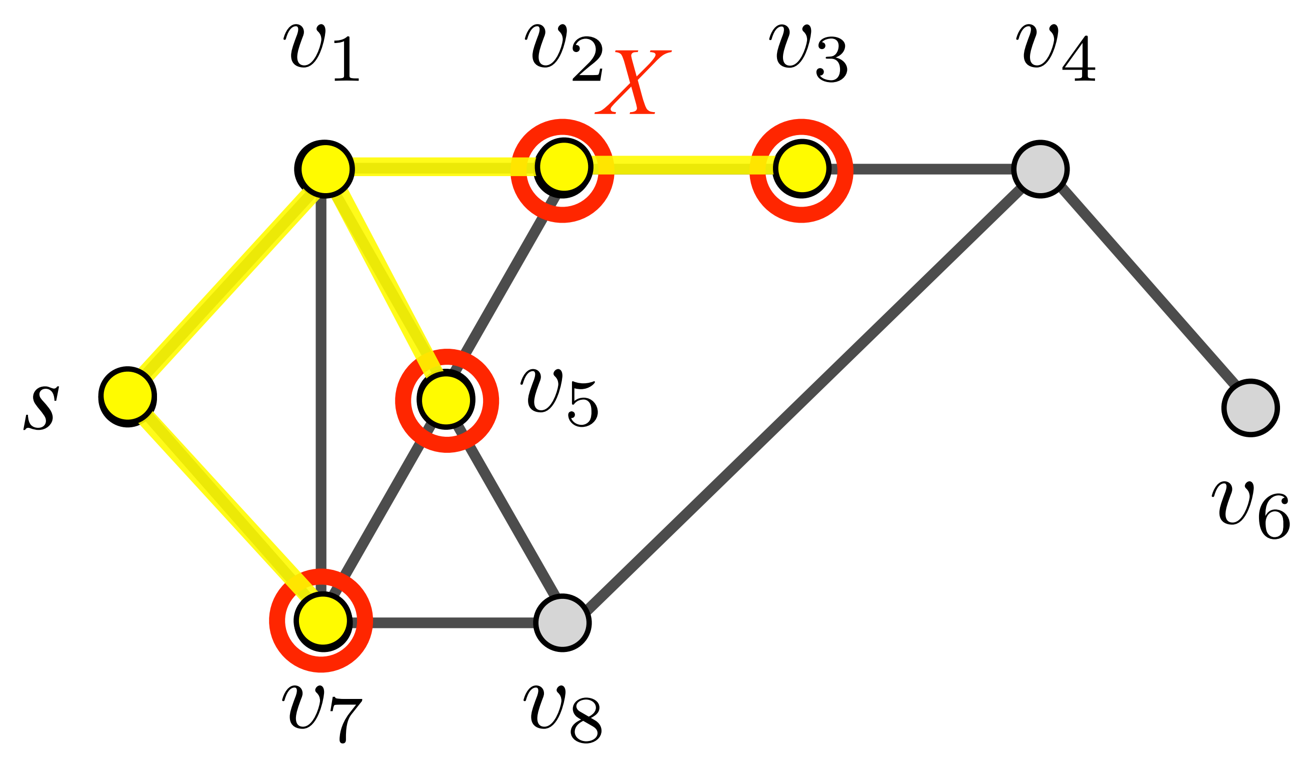
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

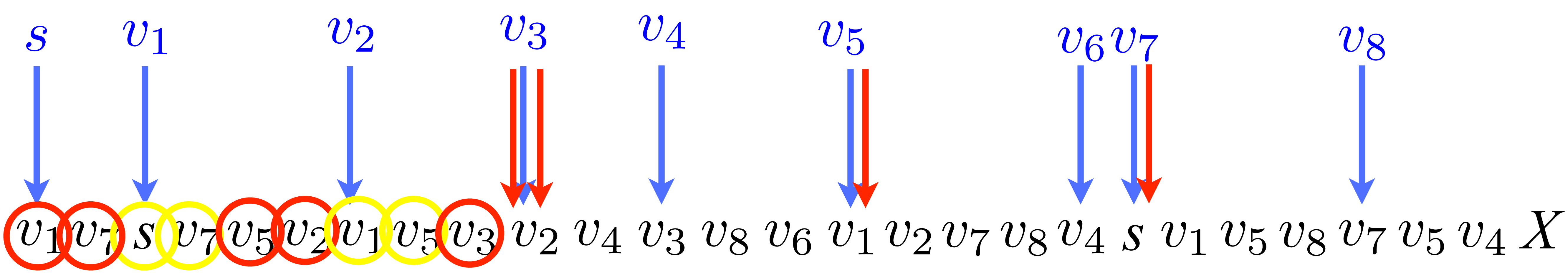
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

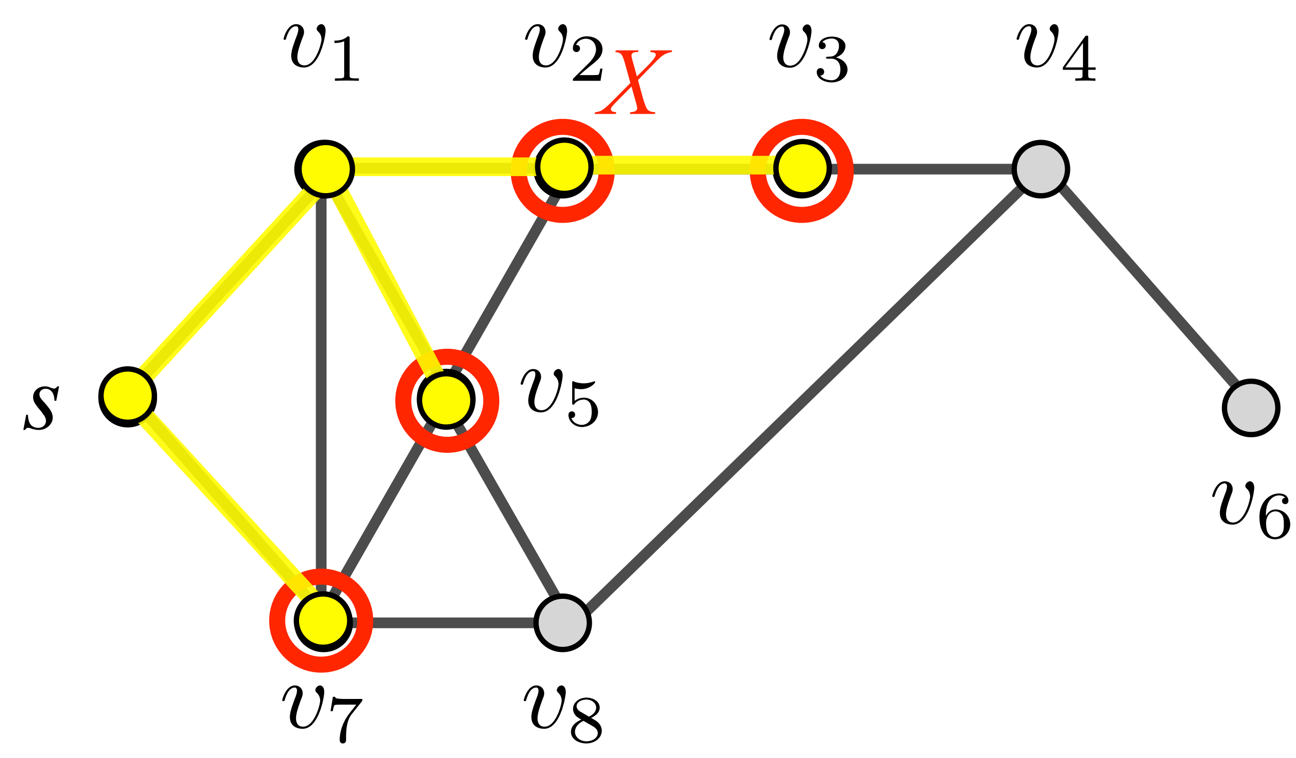
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

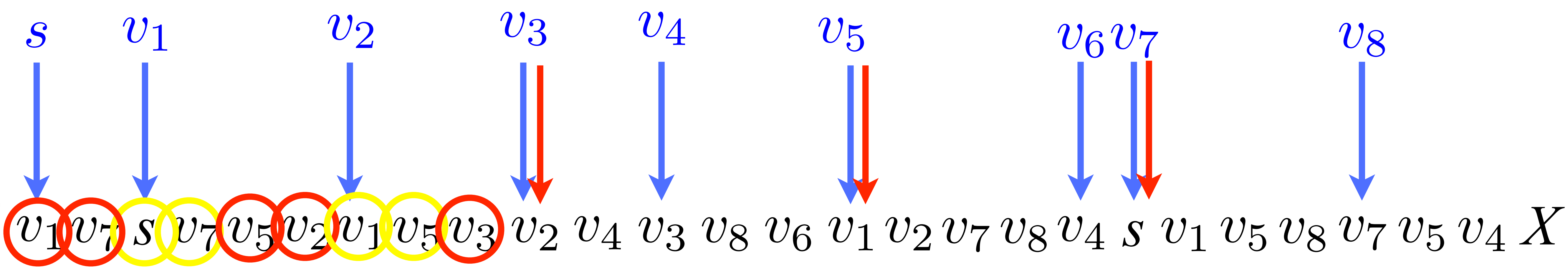
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

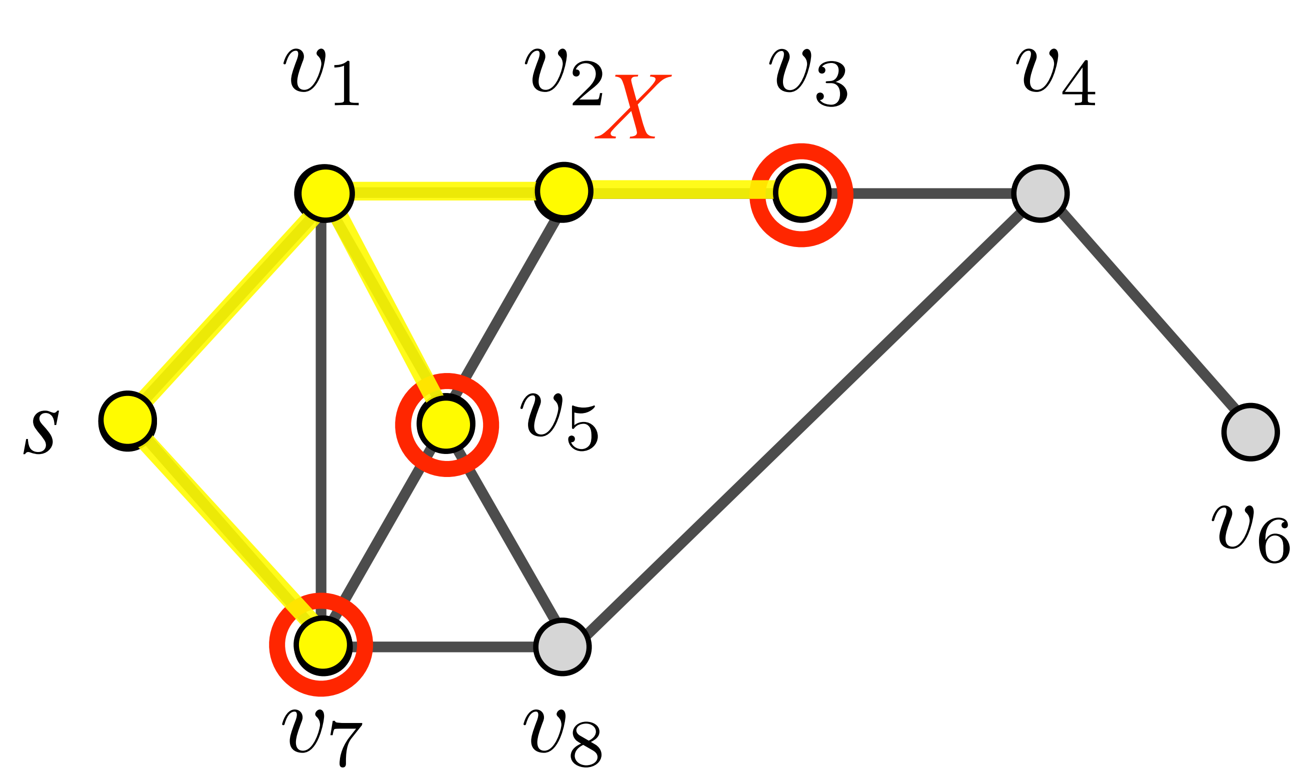
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

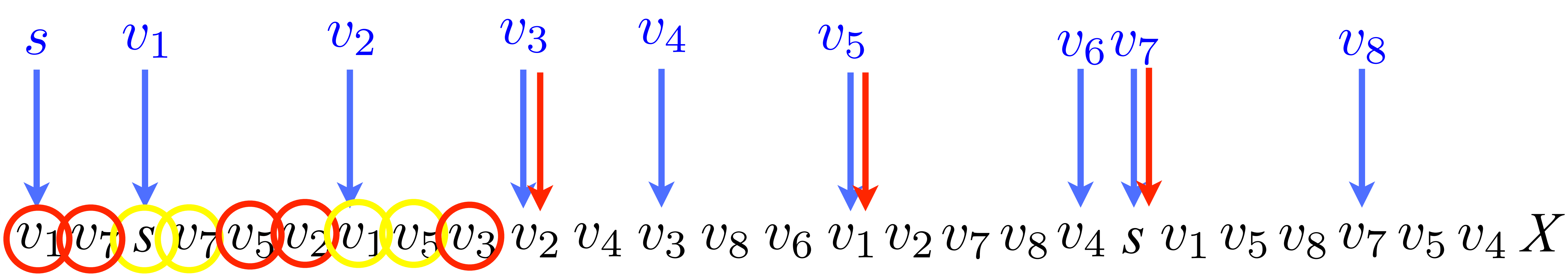
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





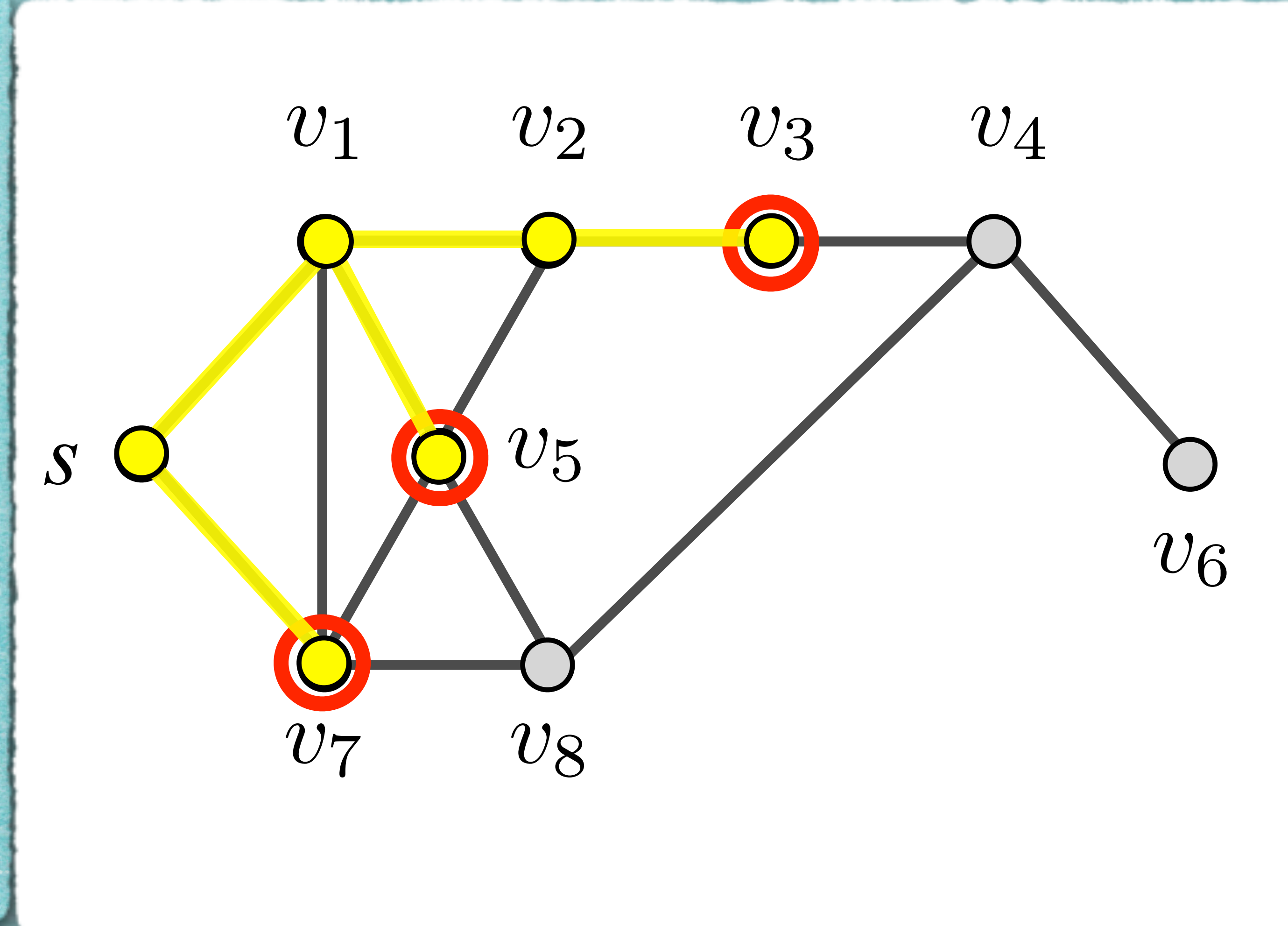
Algorithmus 3.7

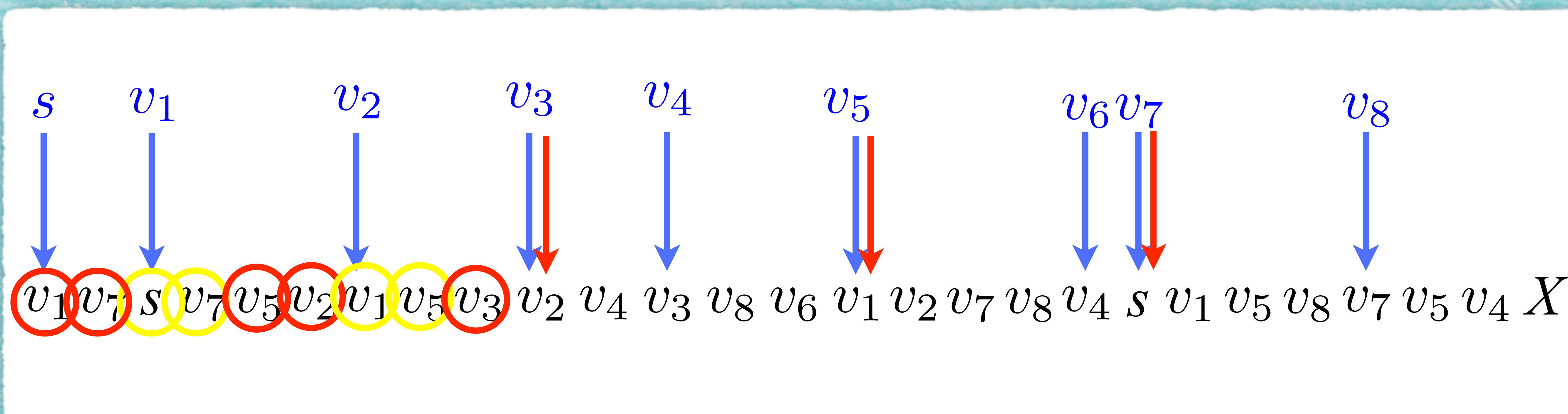
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

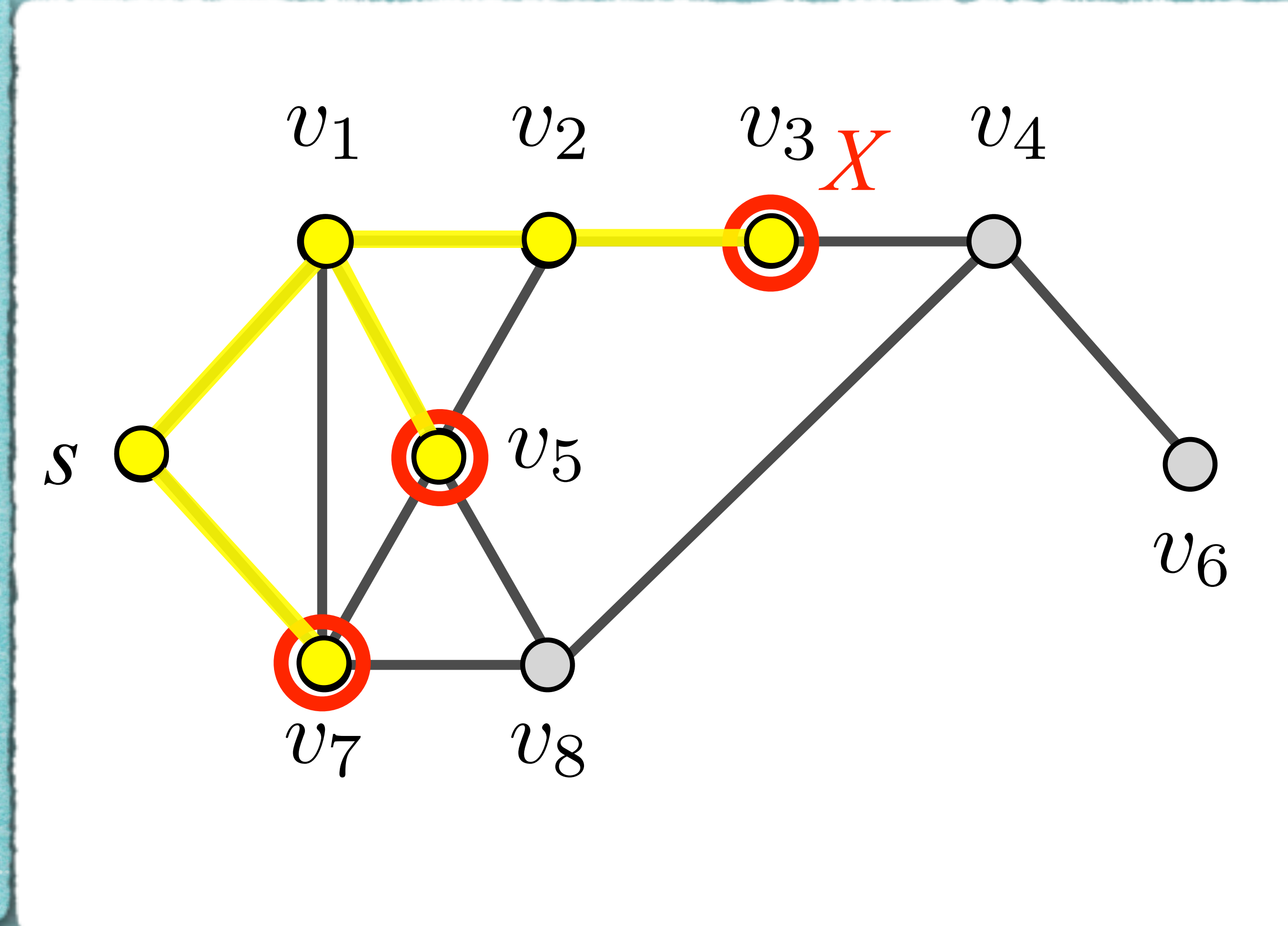
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

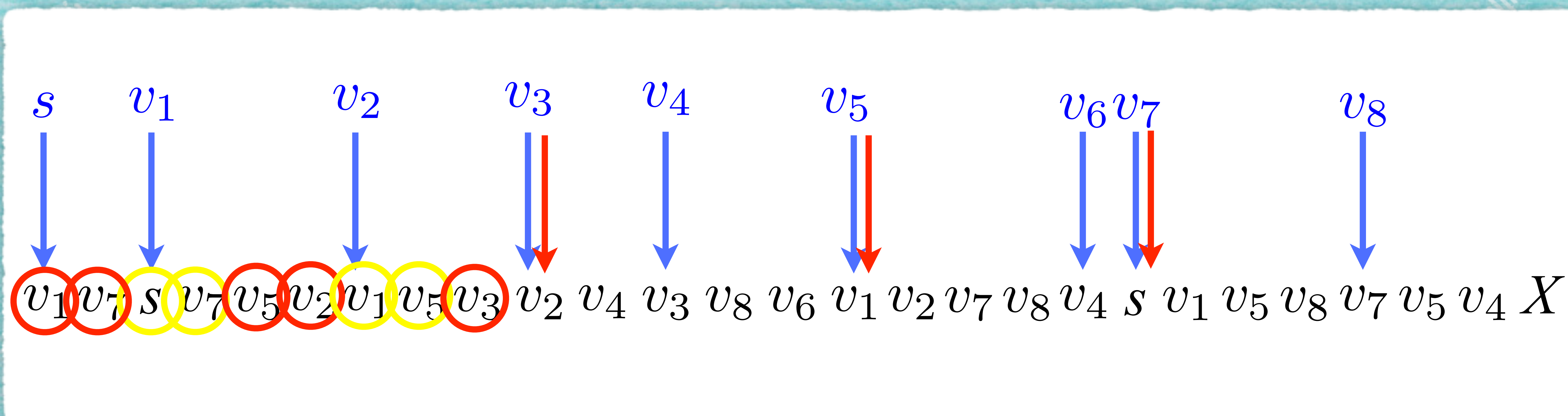
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP



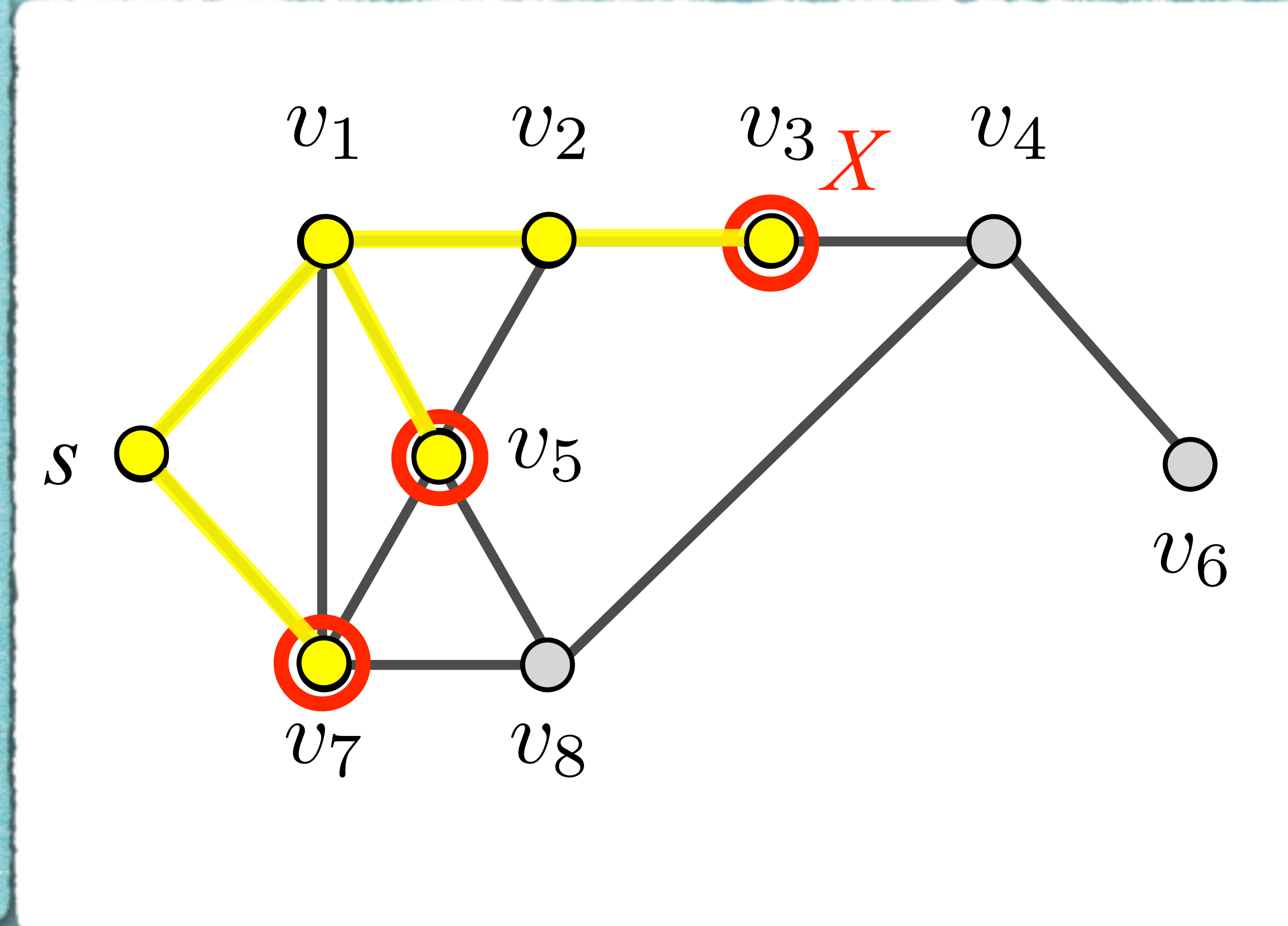


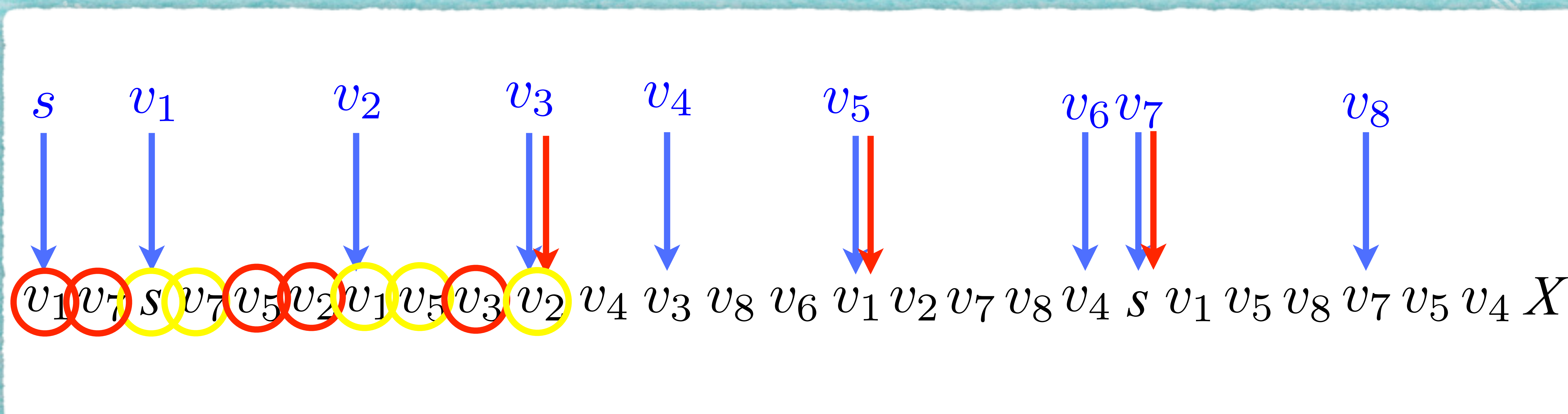
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





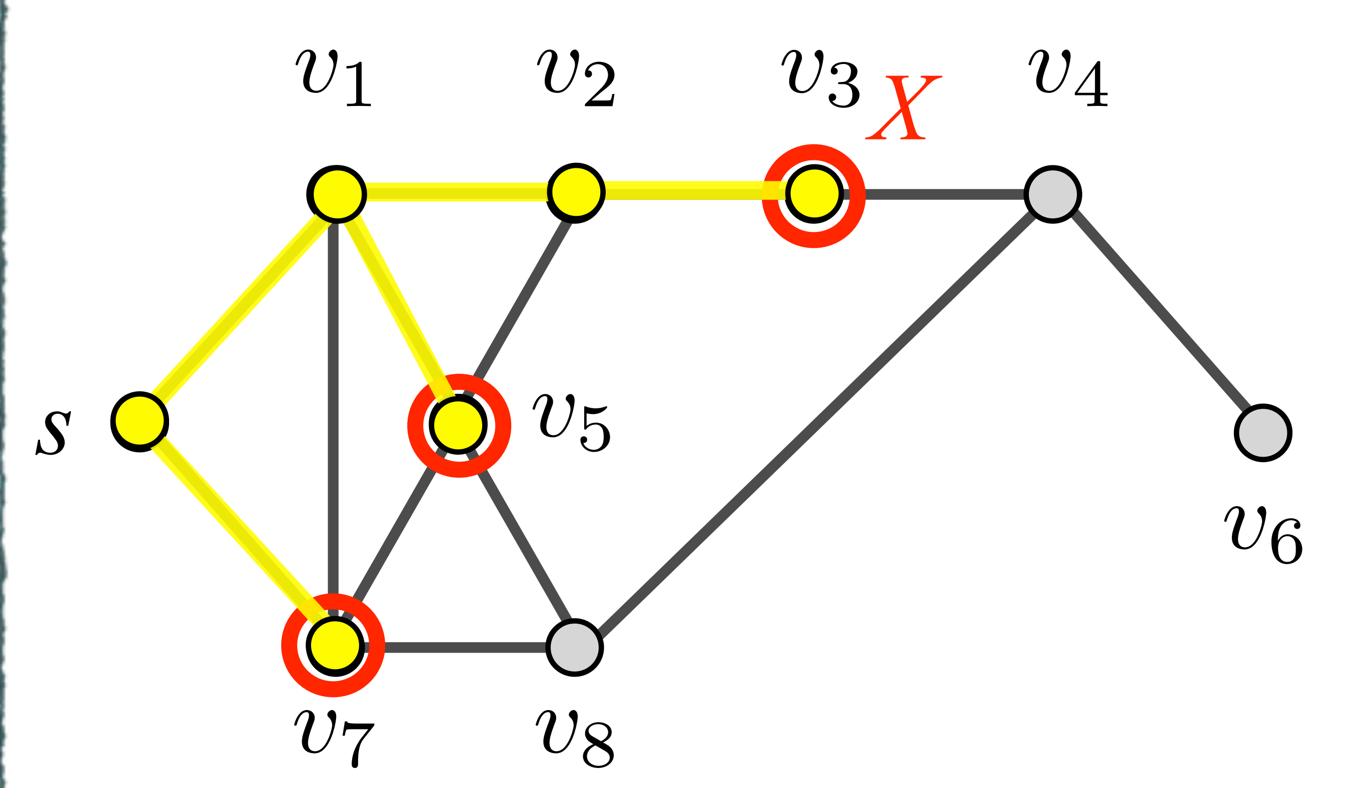
Algorithmus 3.7

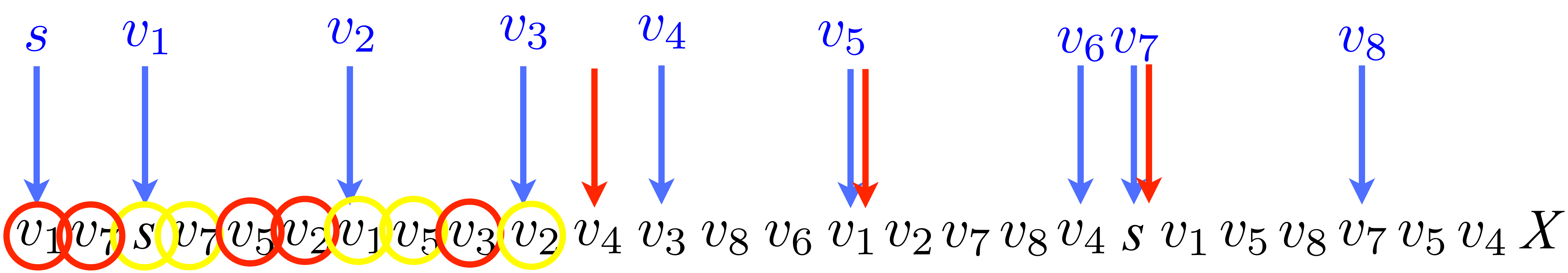
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





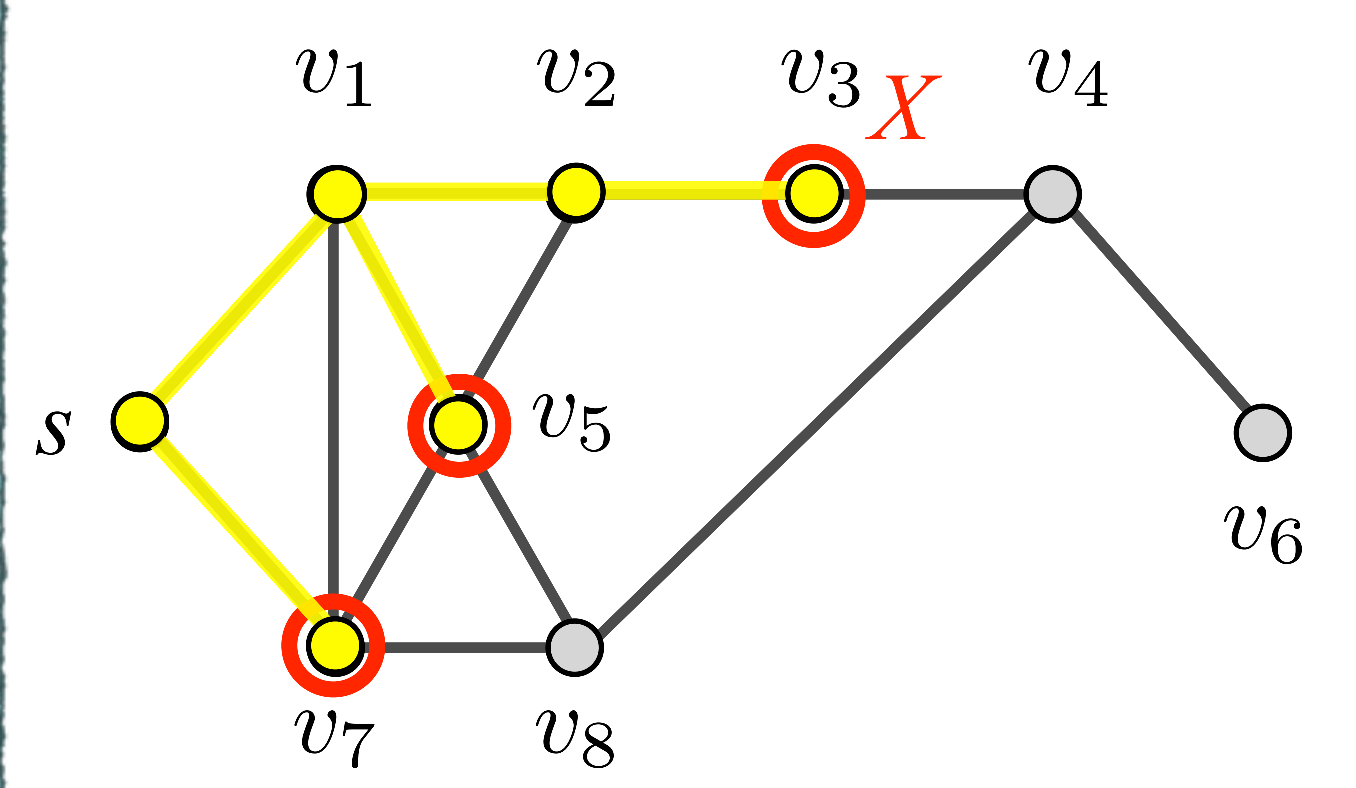
Algorithmus 3.7

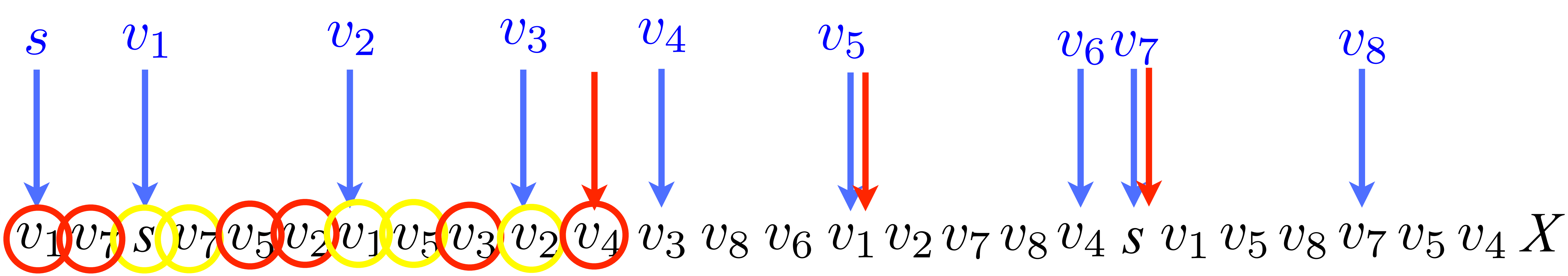
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

    2.2.1. R:=R\{v}

    2.3. ELSE {

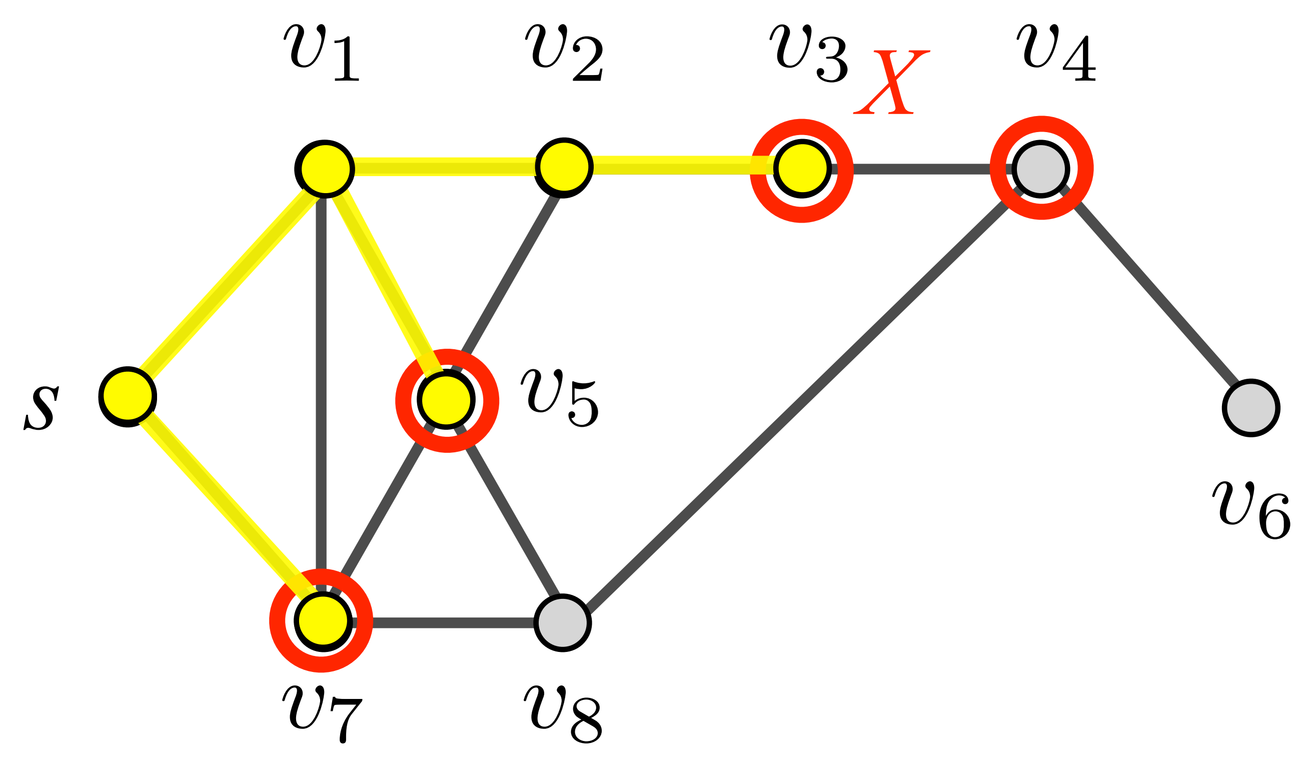
    2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

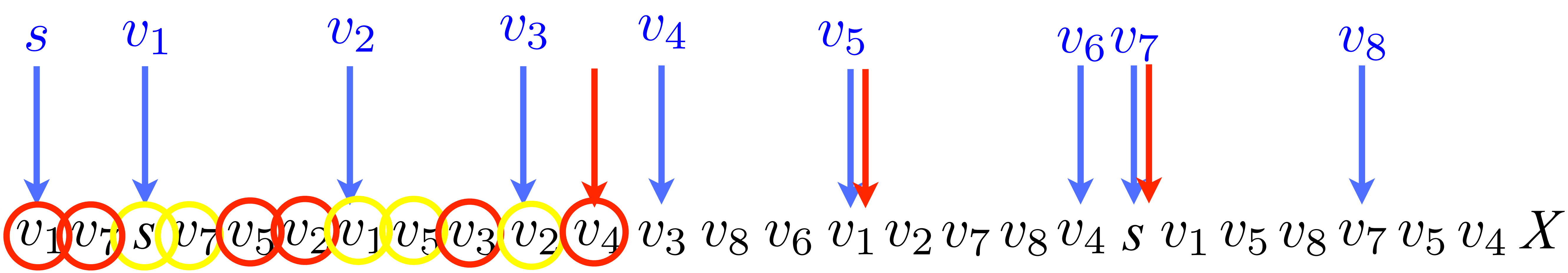
    2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

    }

   }

3. STOP





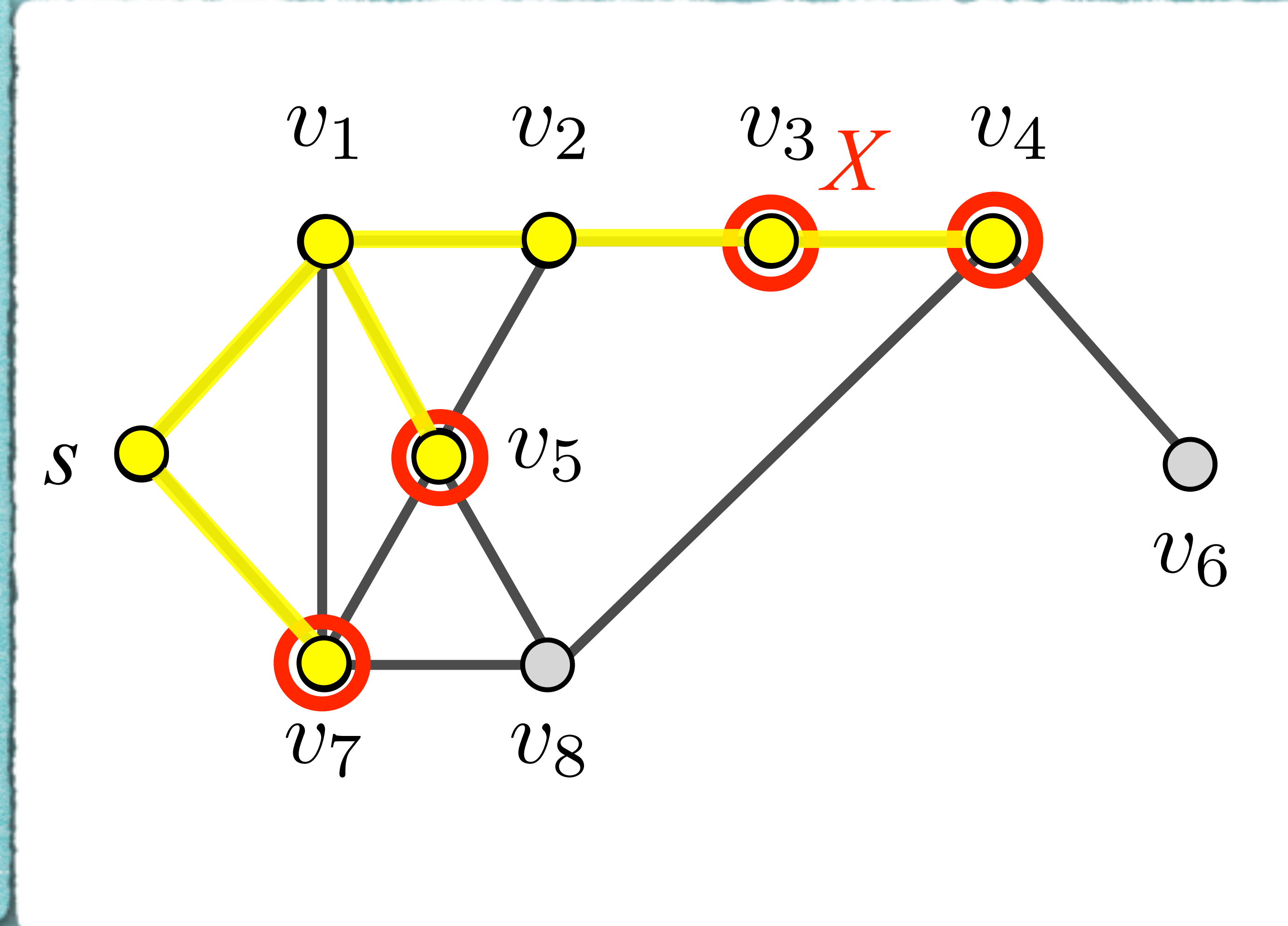
Algorithmus 3.7

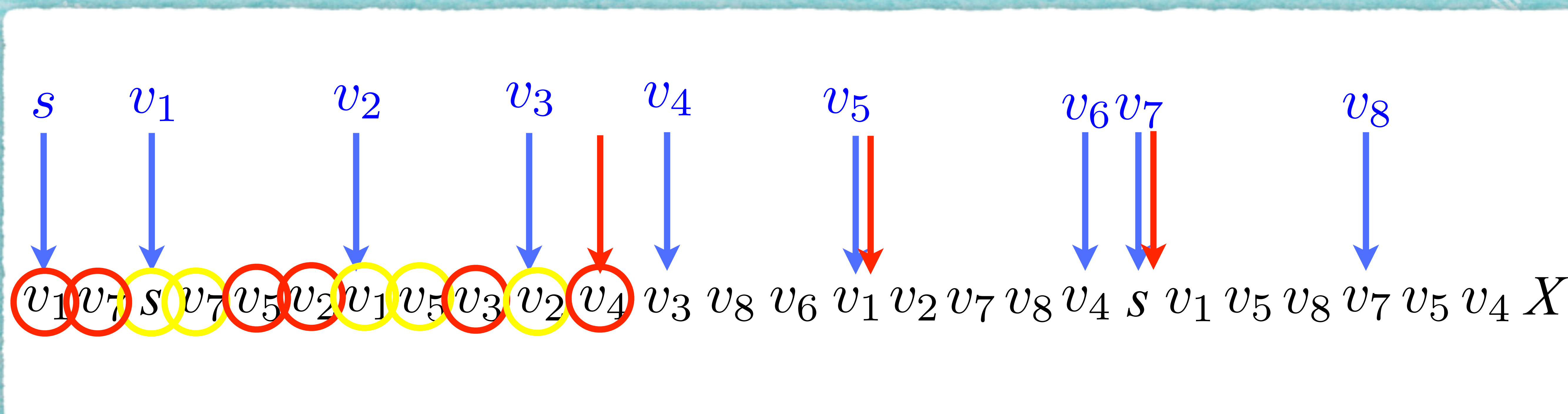
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

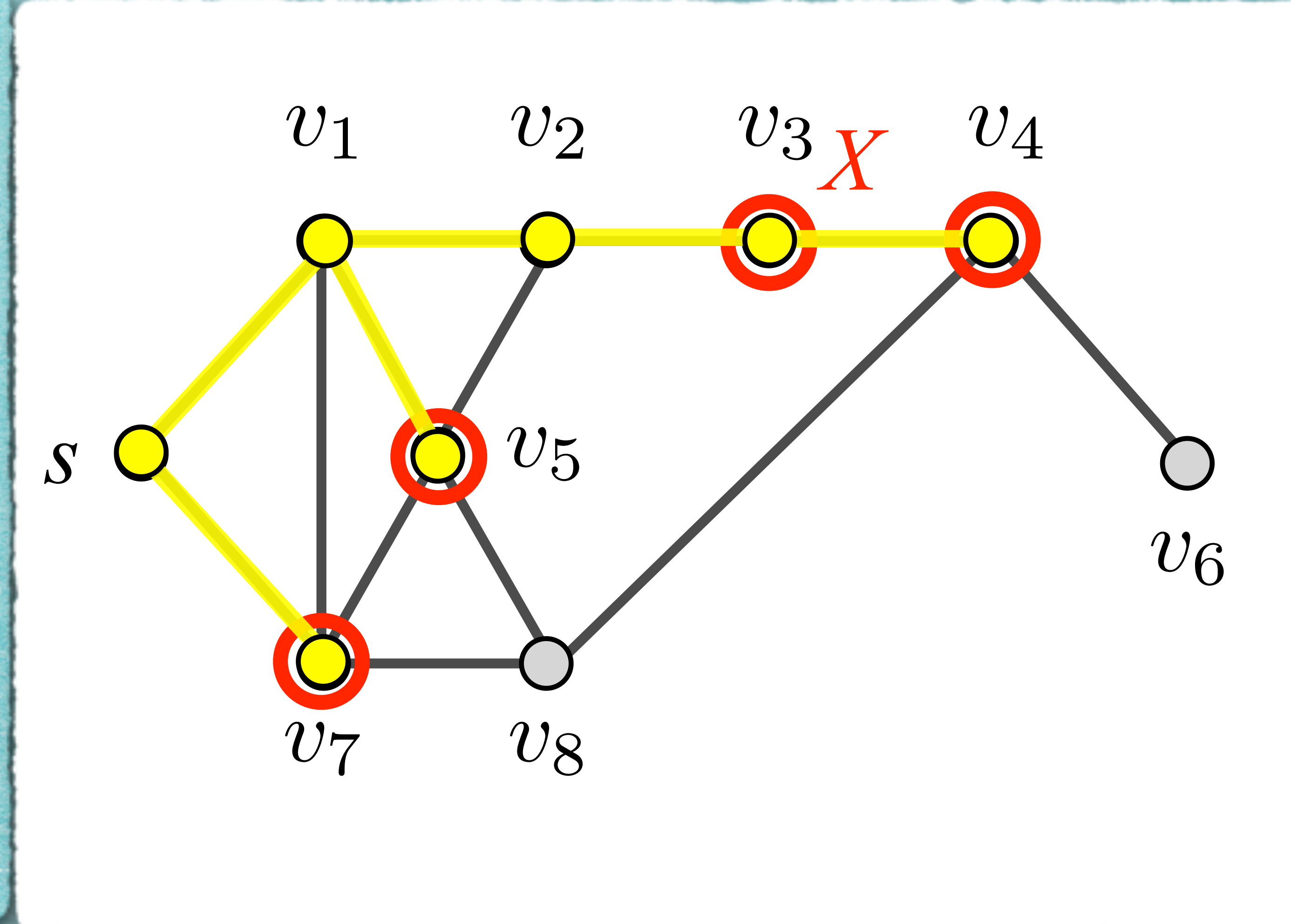
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

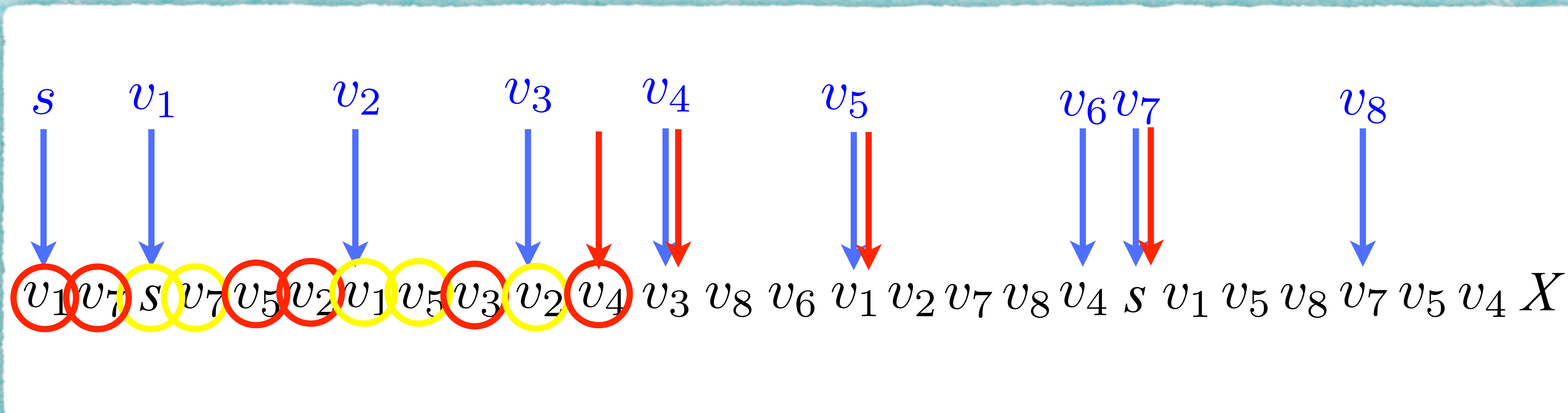
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

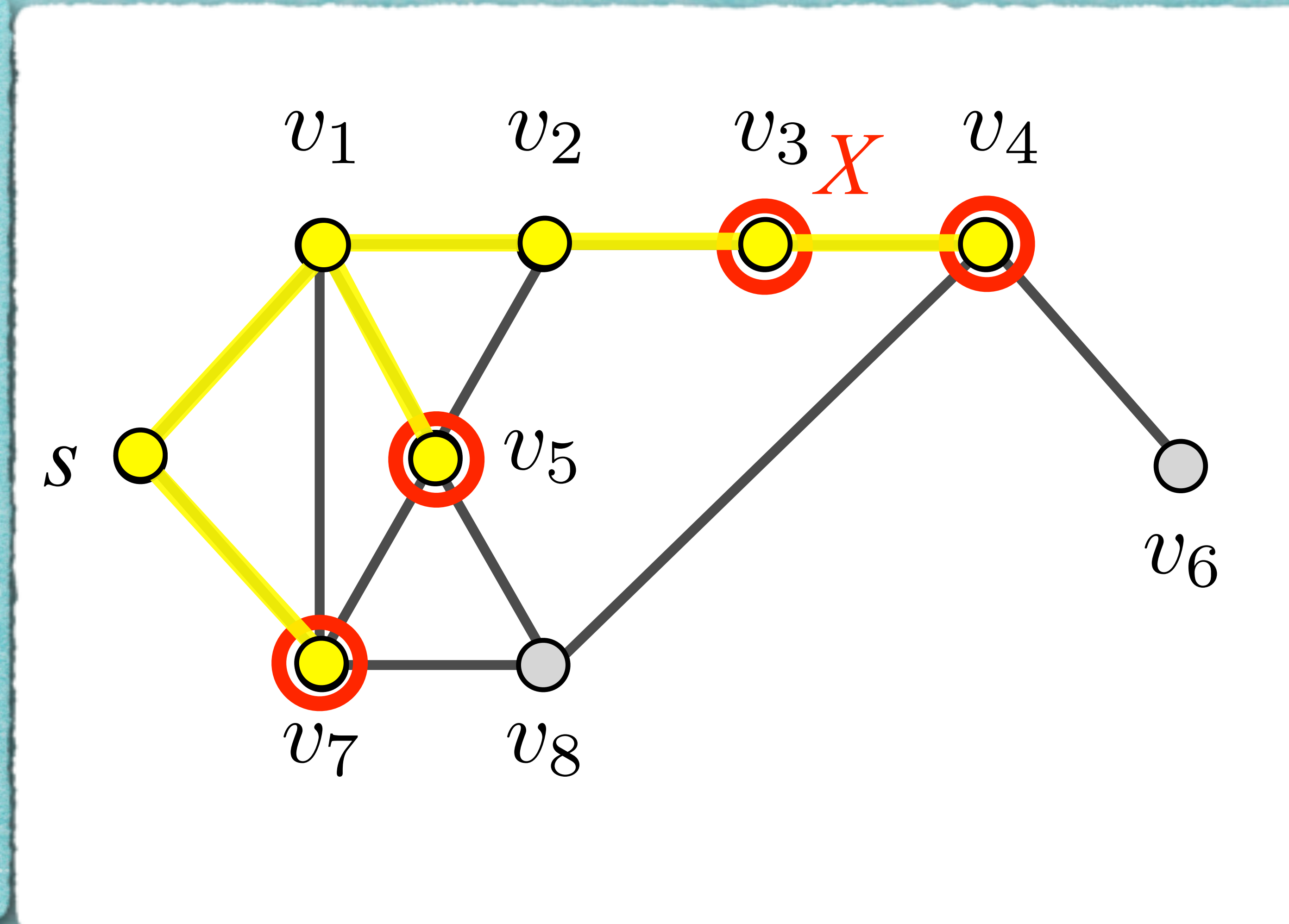
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

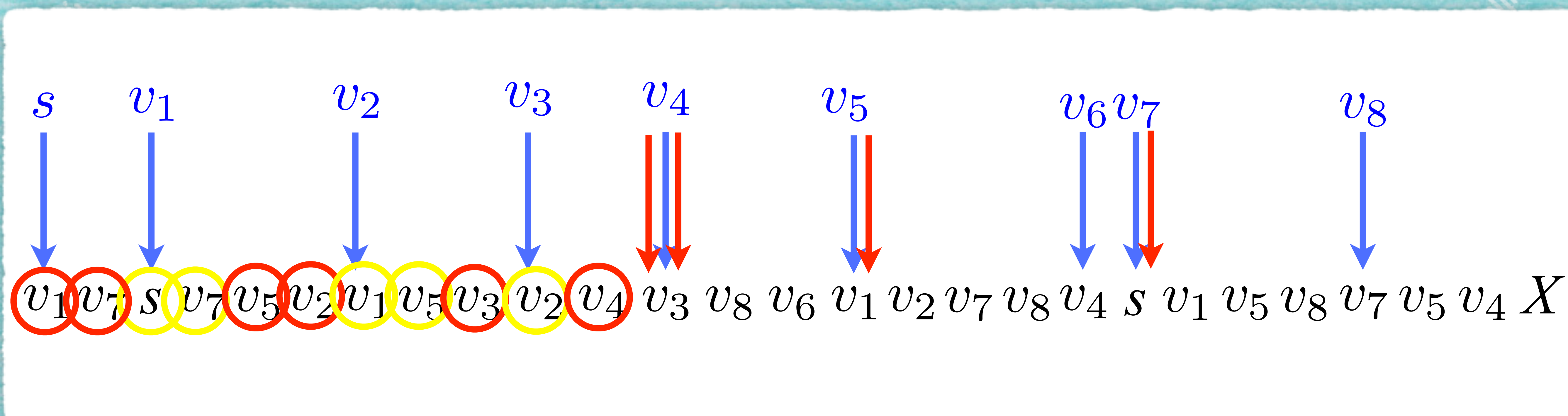
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

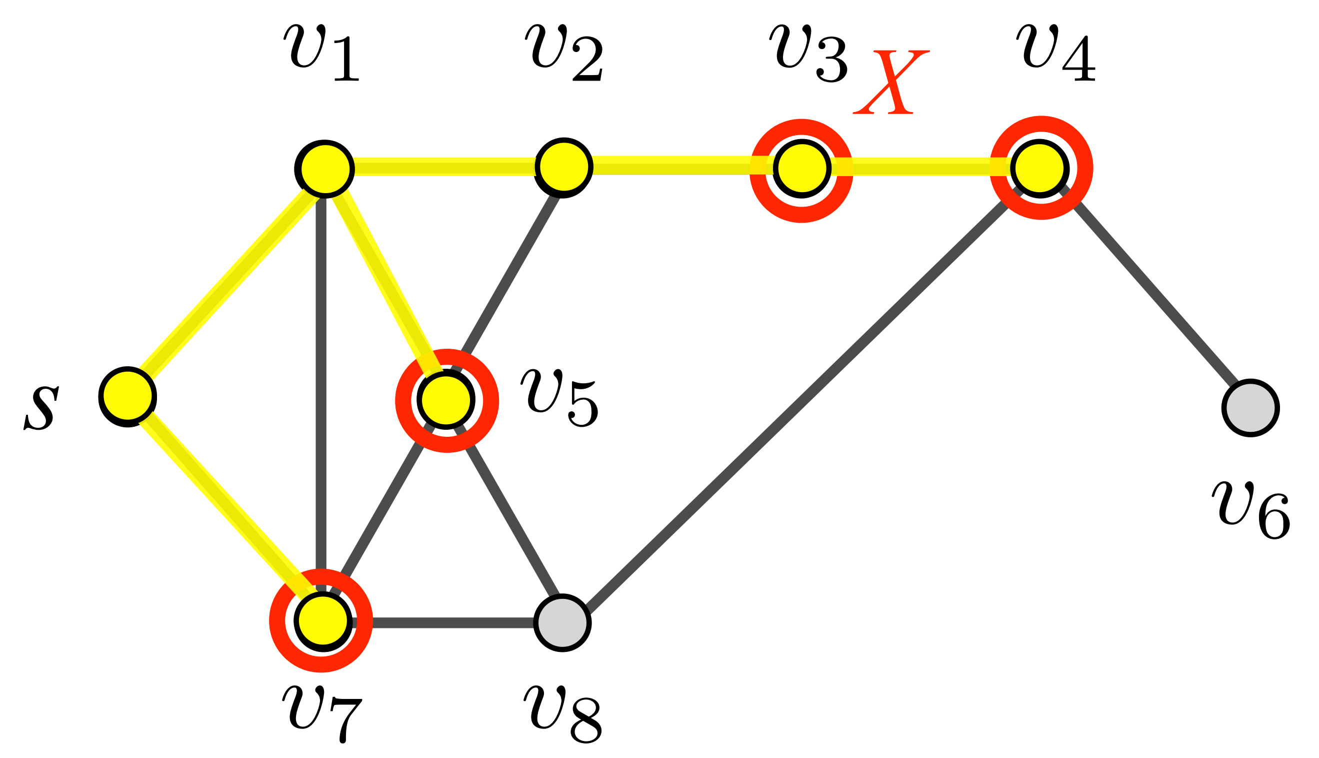
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

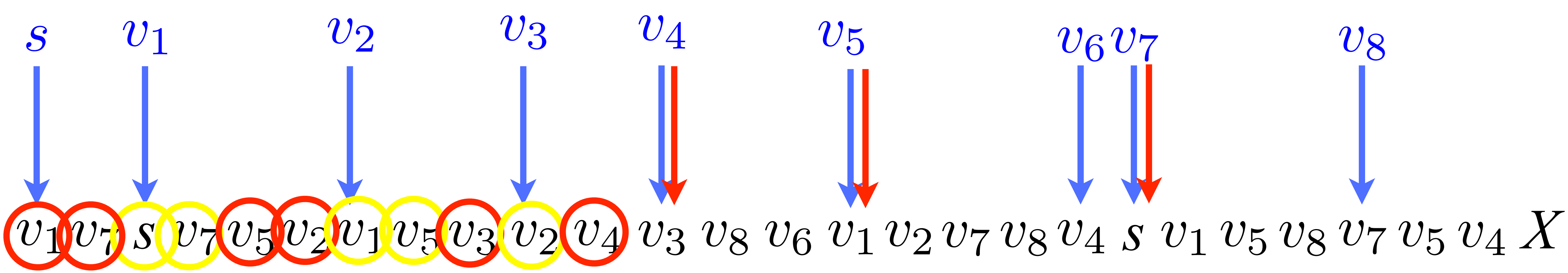
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

2





Algorithmus 3.7

INPUT: Graph $G = (V,E)$, Knoten s

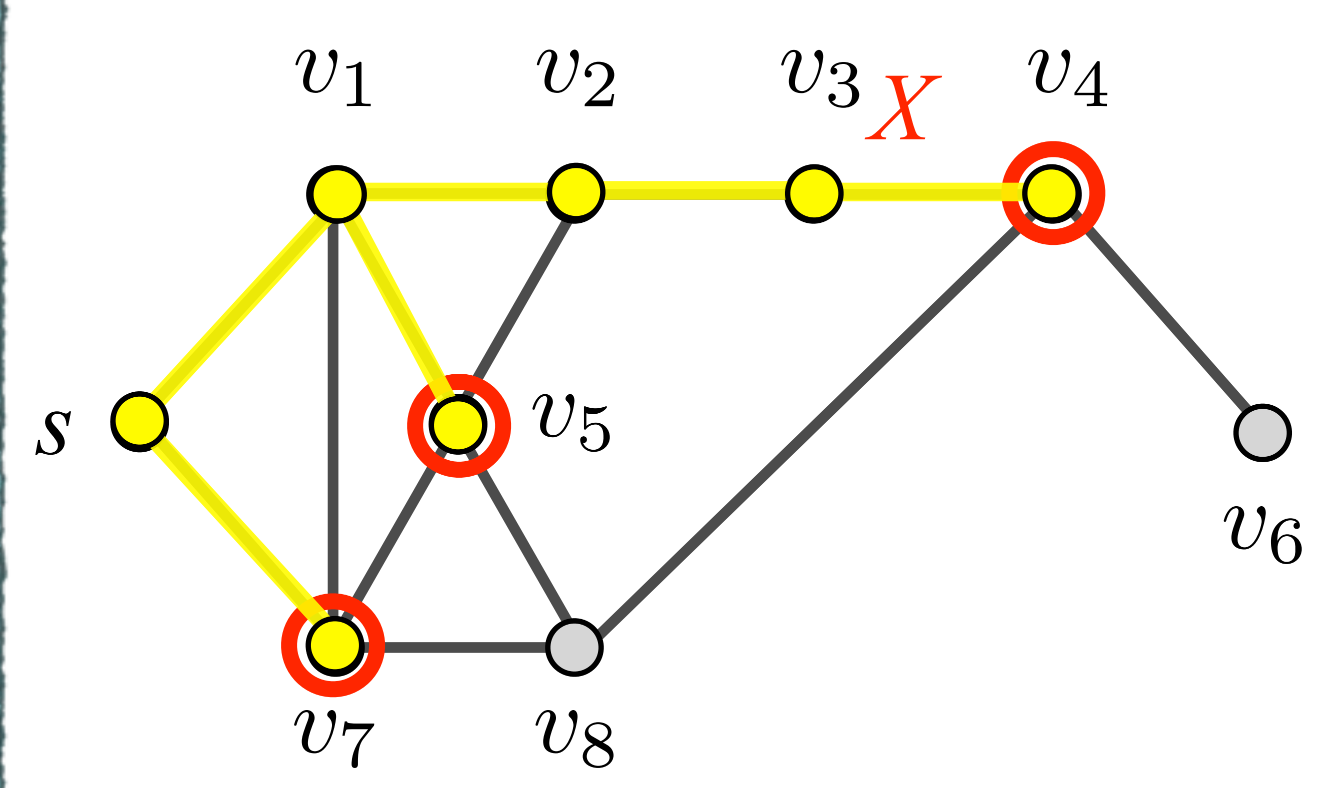
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

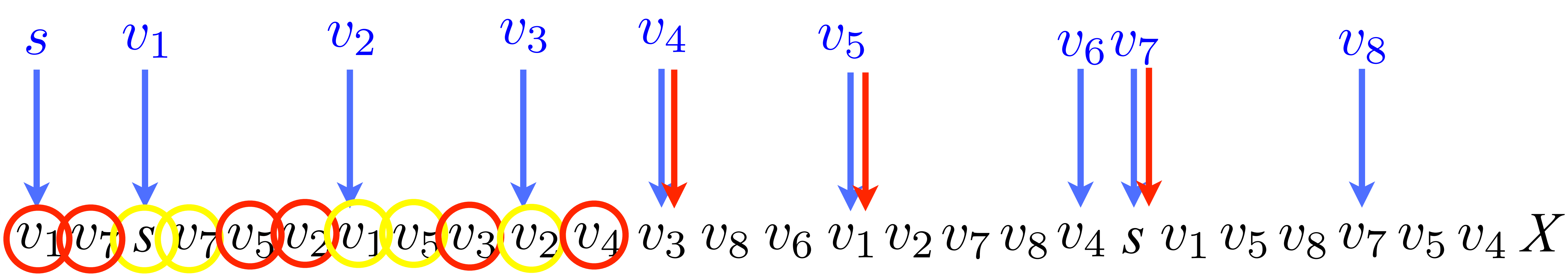
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R:=R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

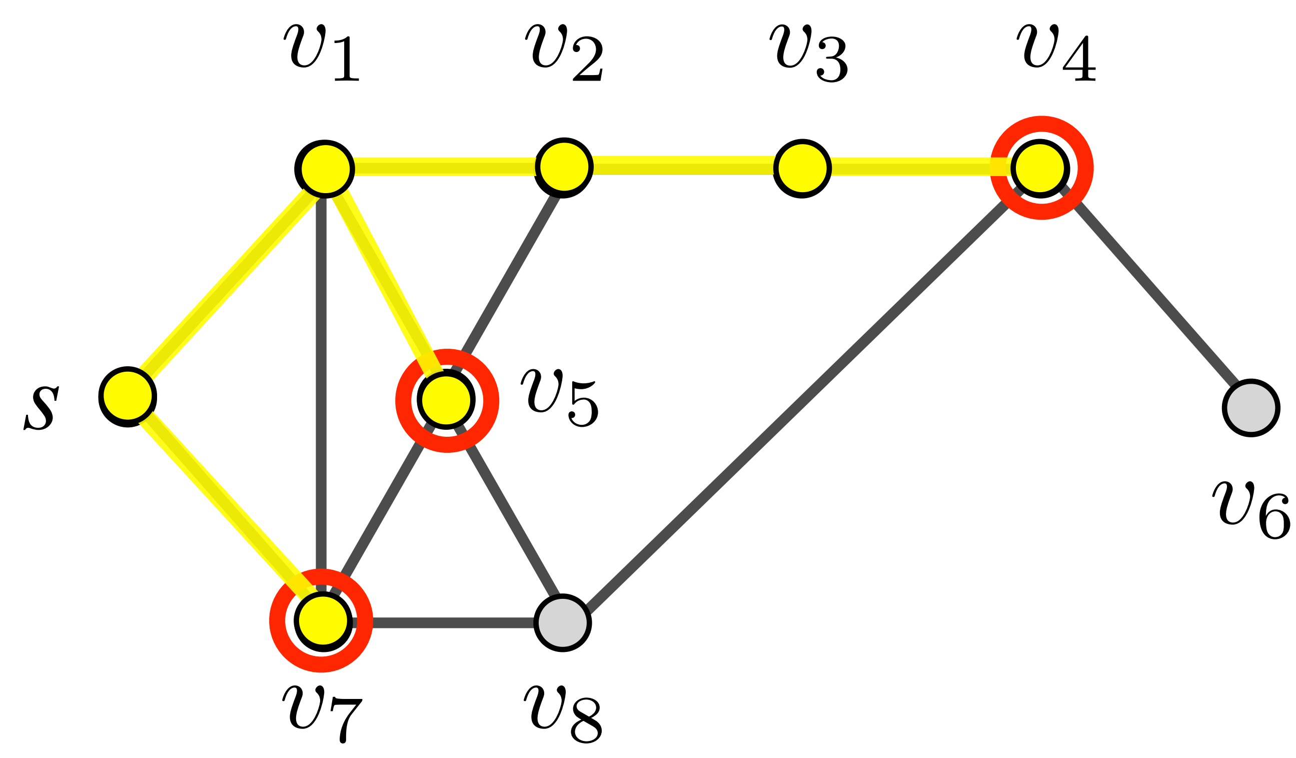
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

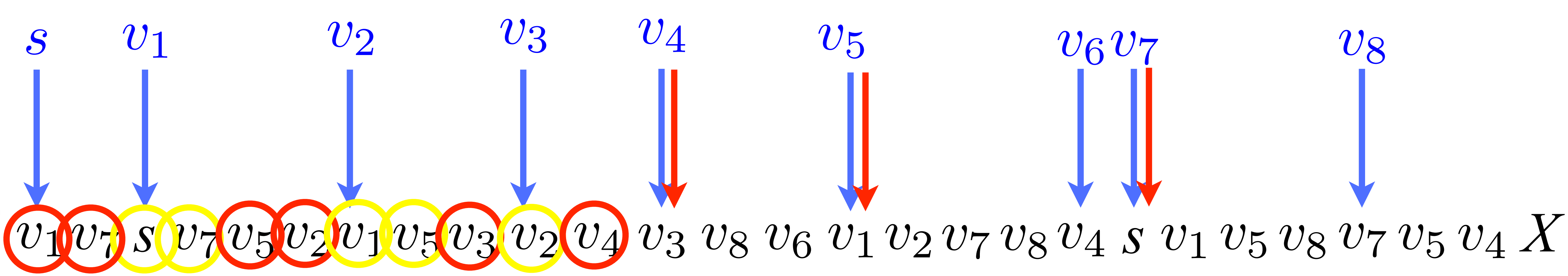
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP



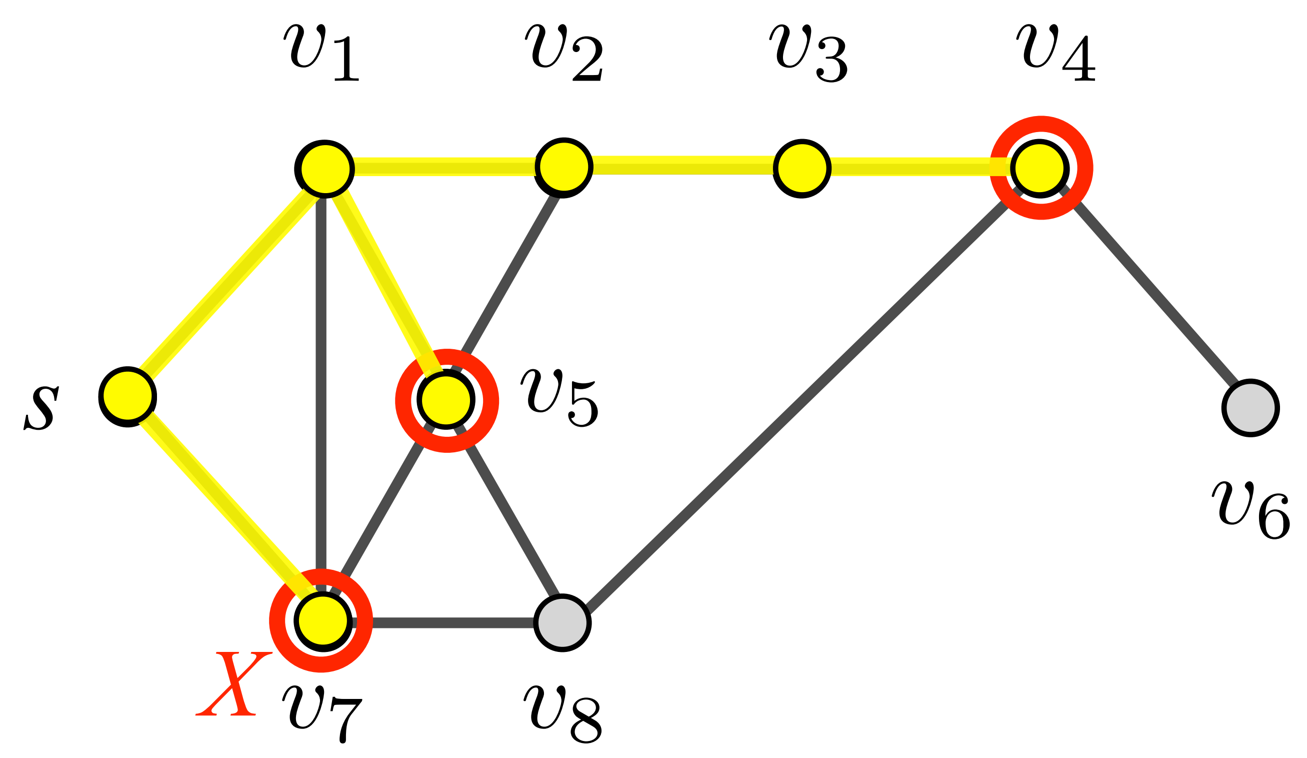


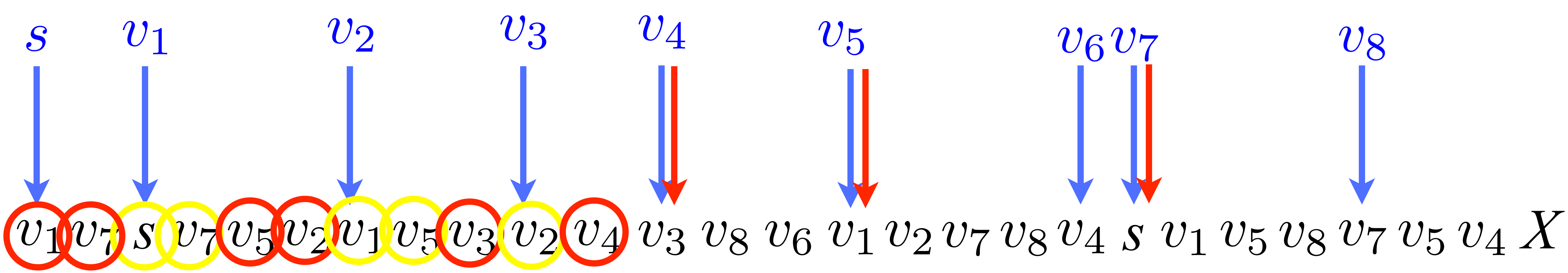
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





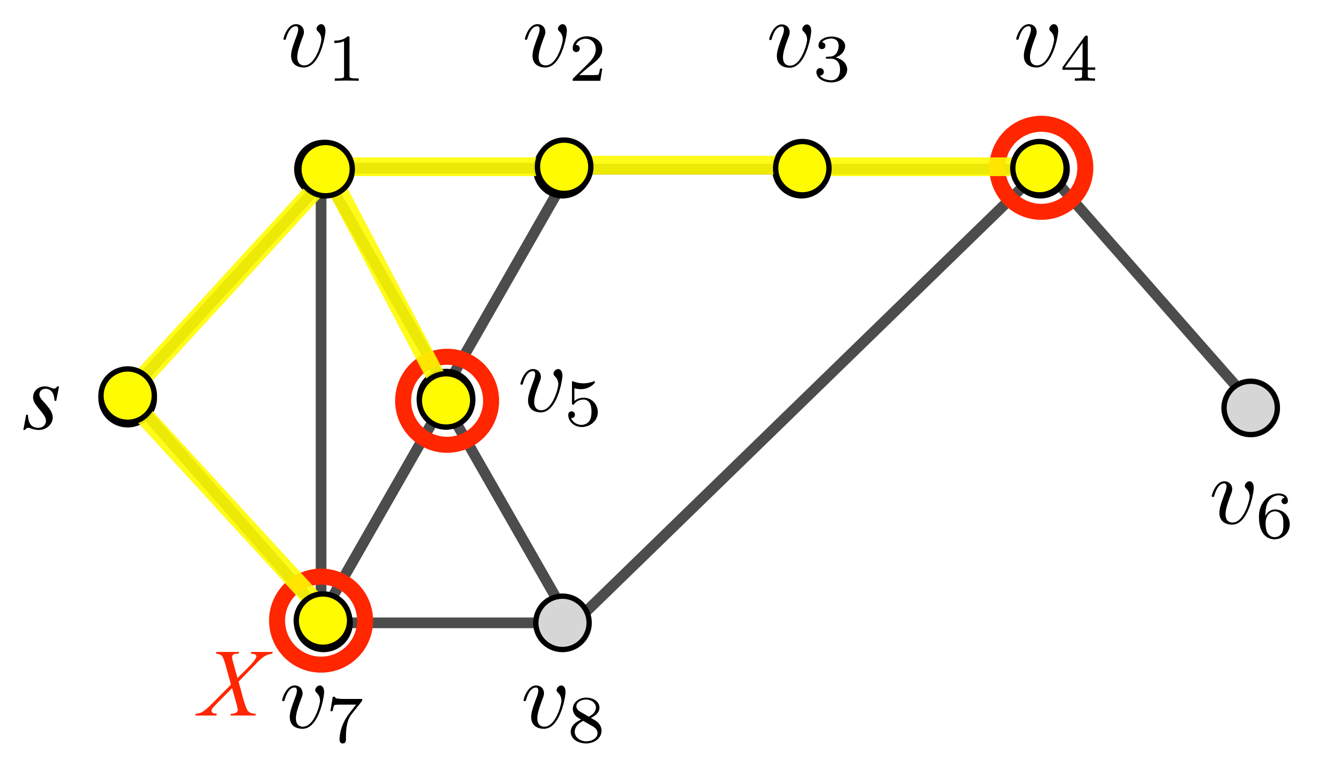
Algorithmus 3.7

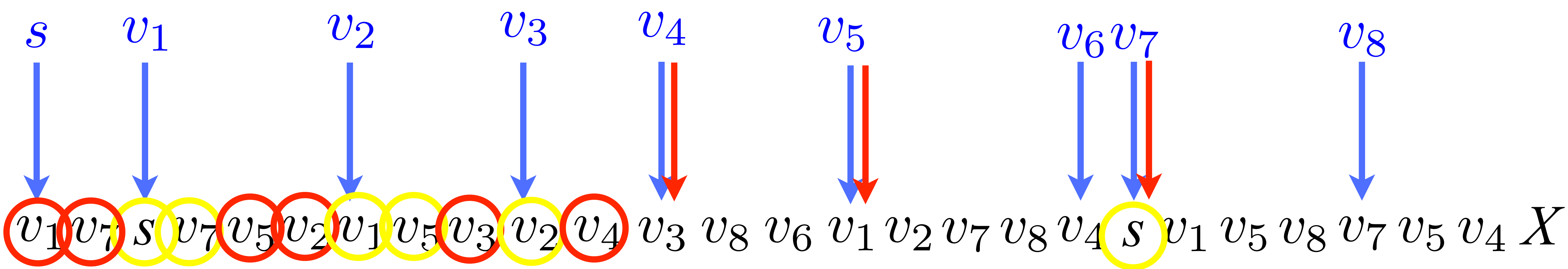
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }
    - }
3. STOP





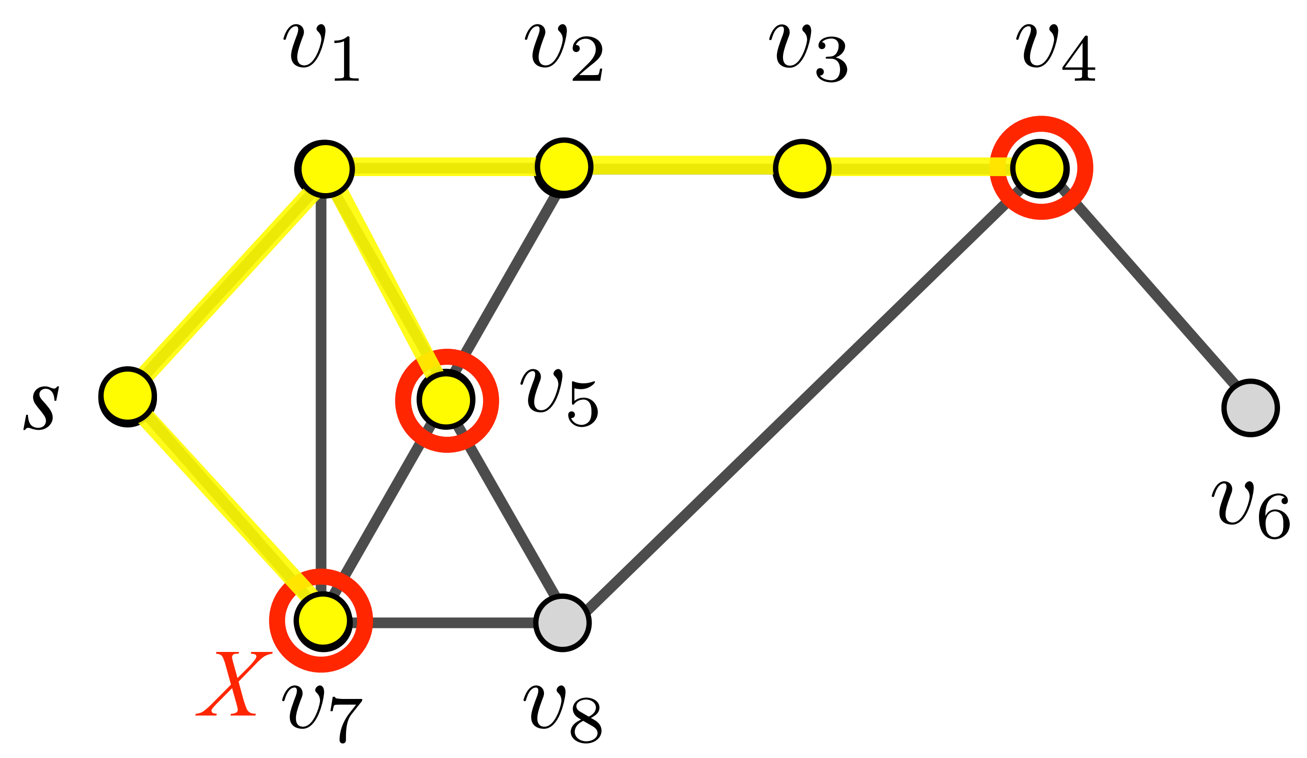
Algorithmus 3.7

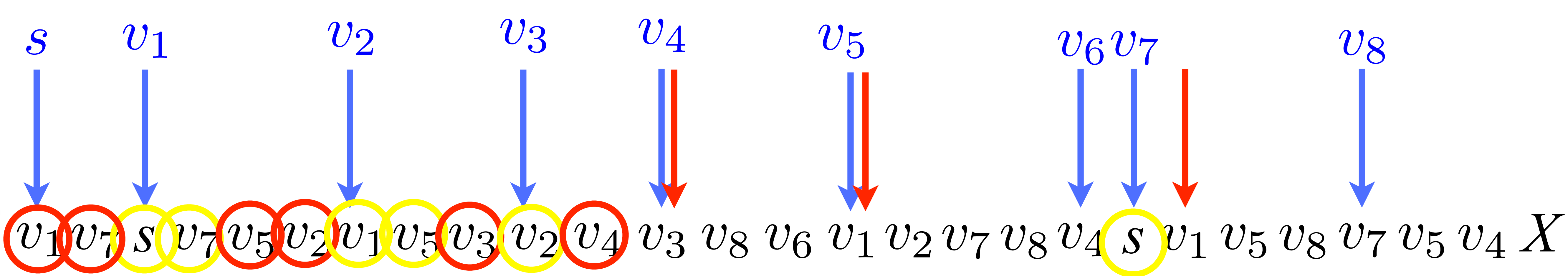
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP

2





Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

    2.1. Wähle v∈ R

    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

    2.2.1. R:=R\{v}

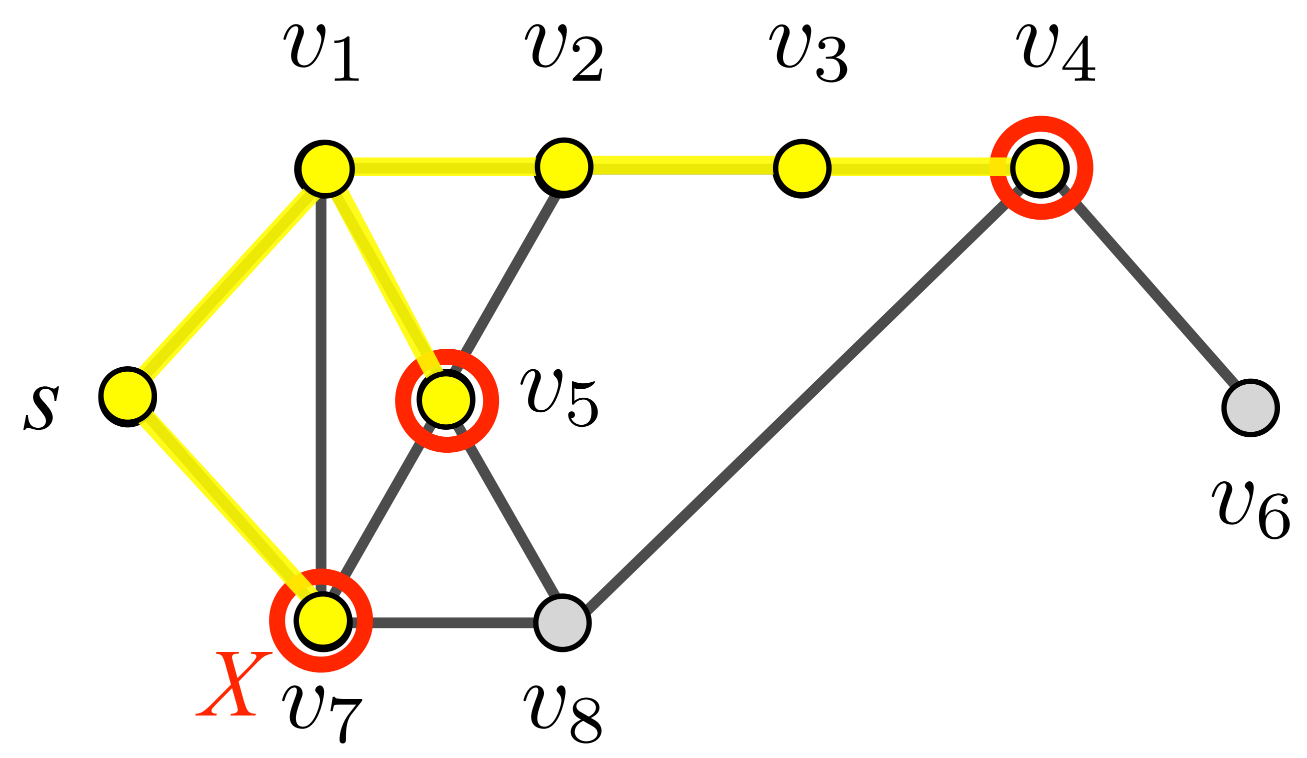
    2.3. ELSE {

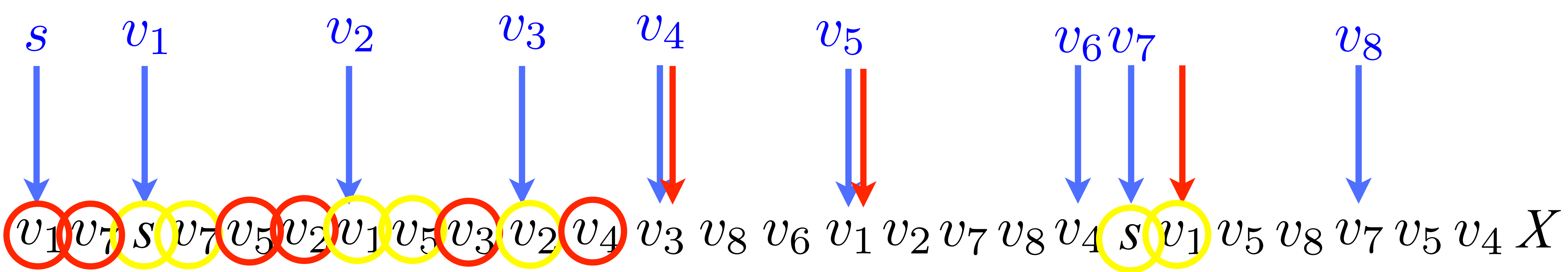
    2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E

    2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

    }

   }
3. STOP

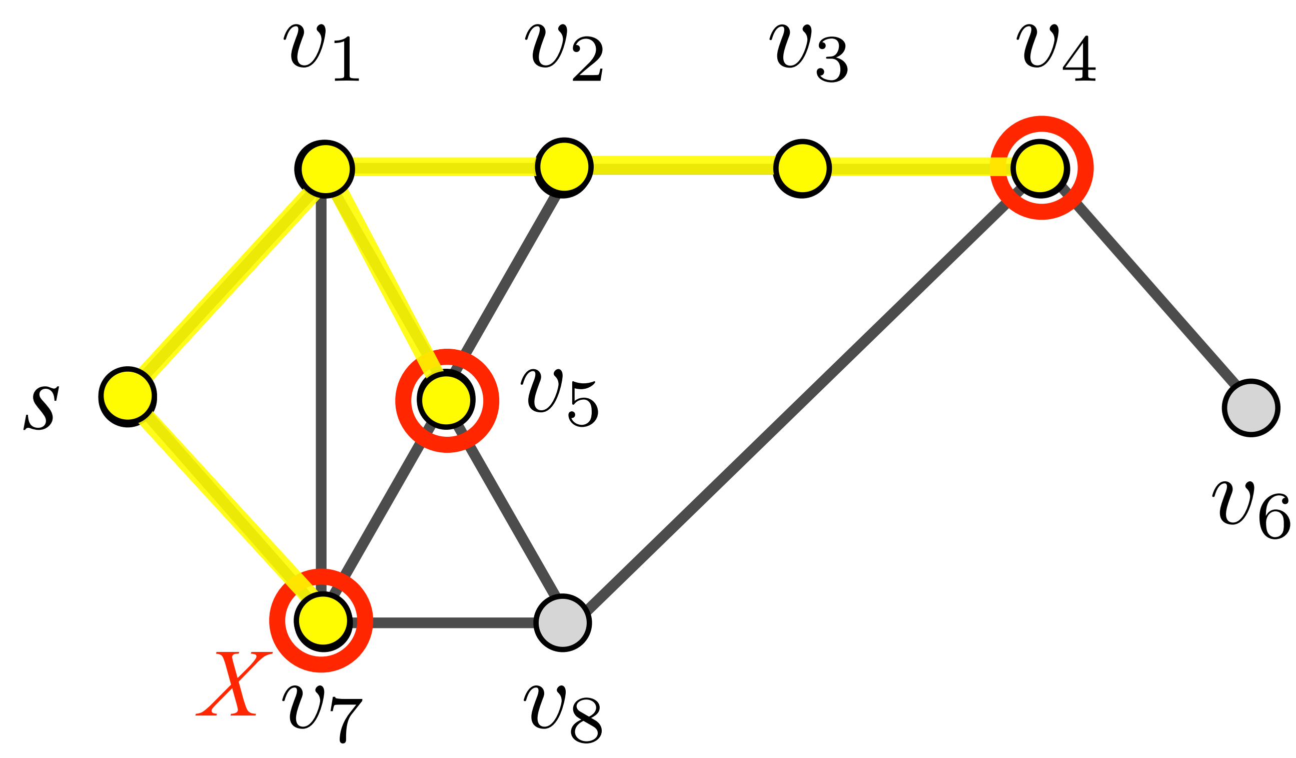


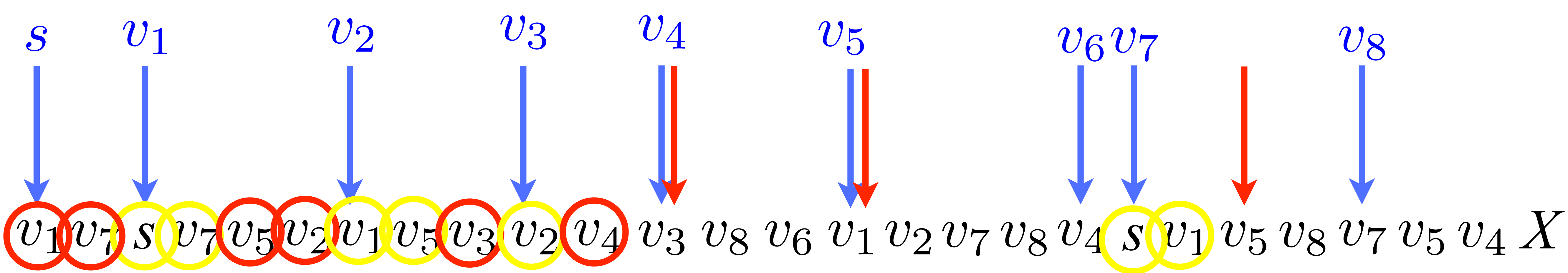


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }
    - }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

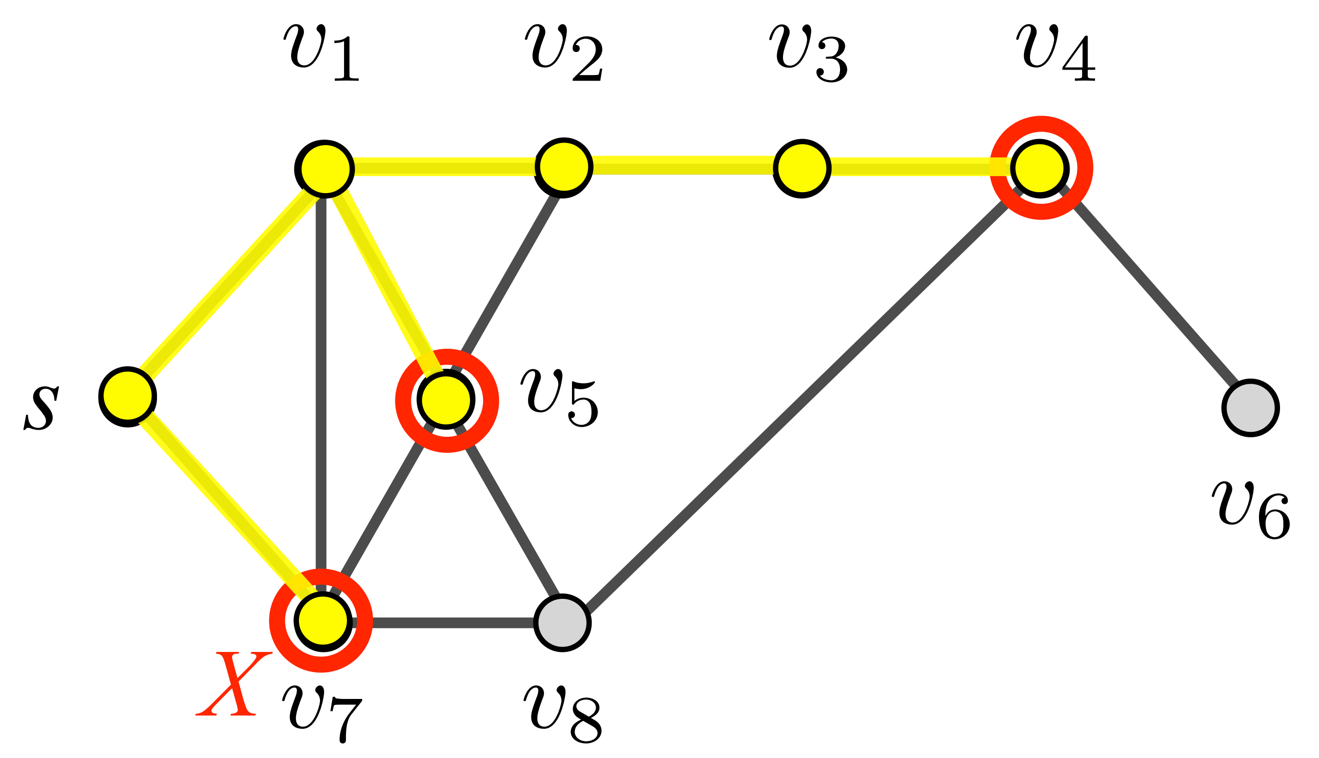
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

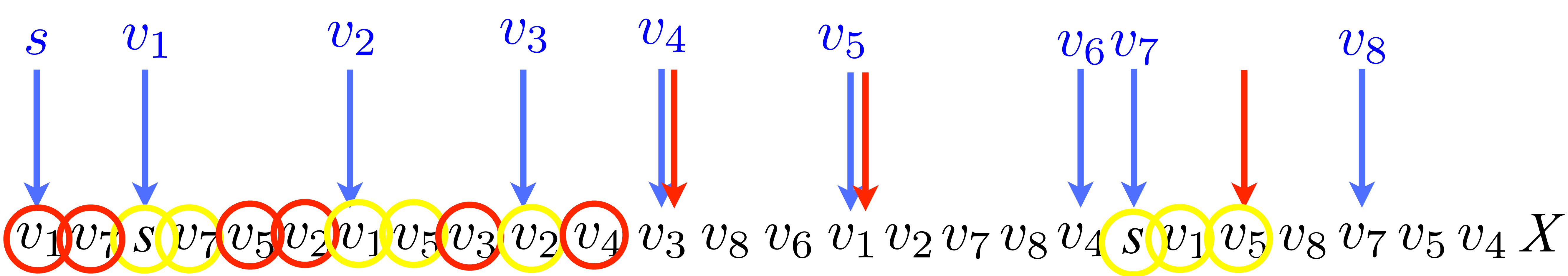
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

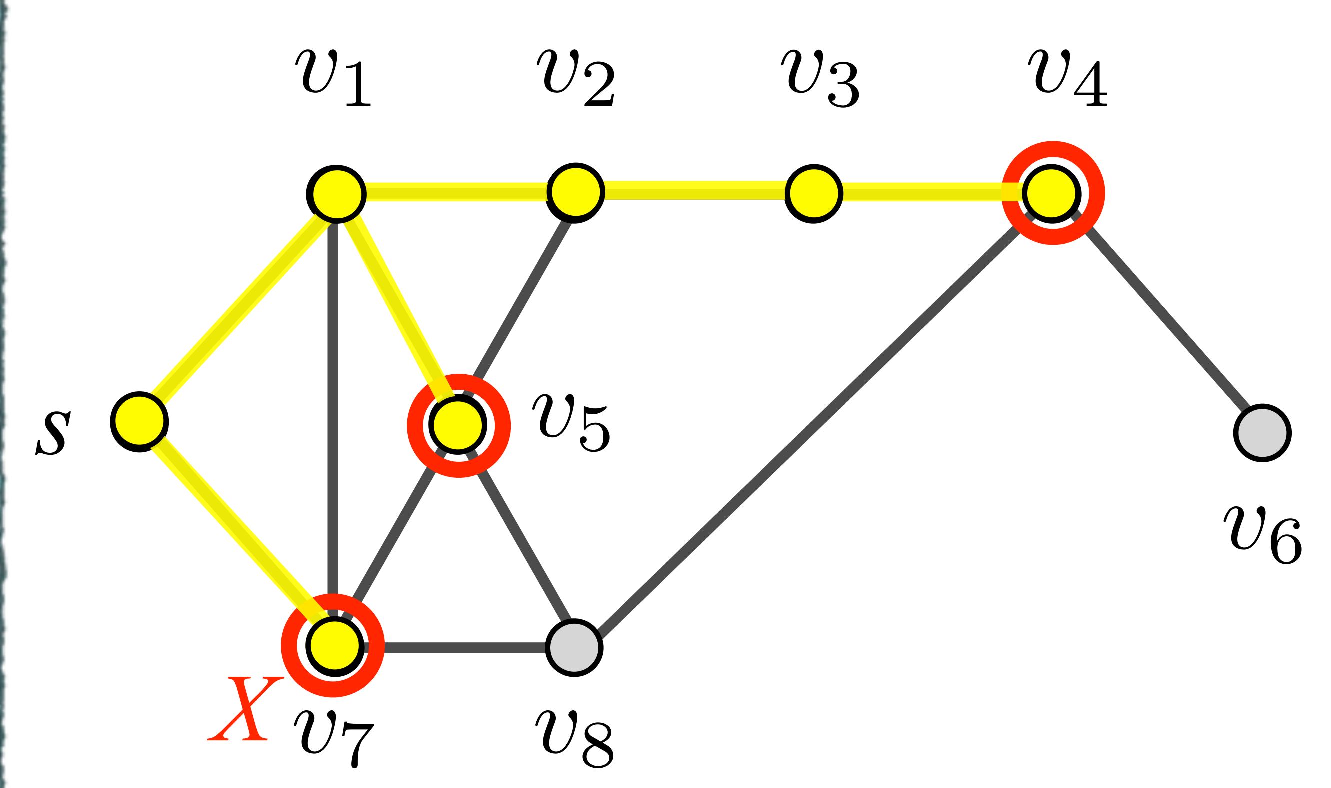
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

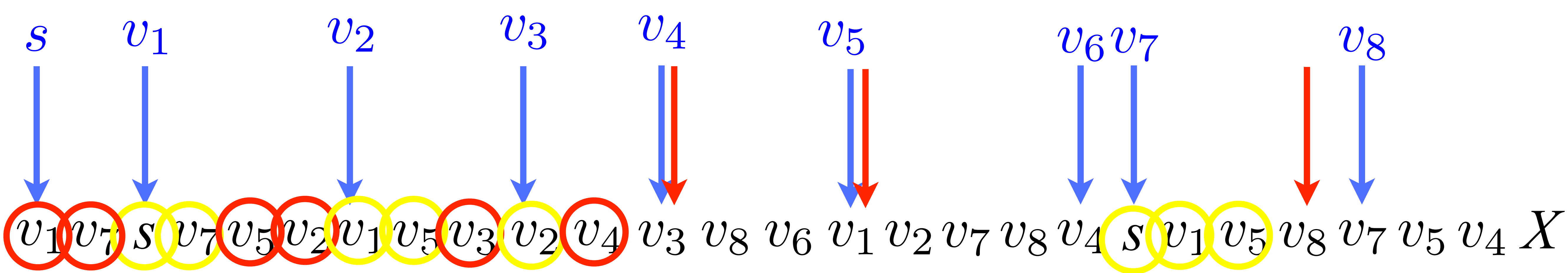
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

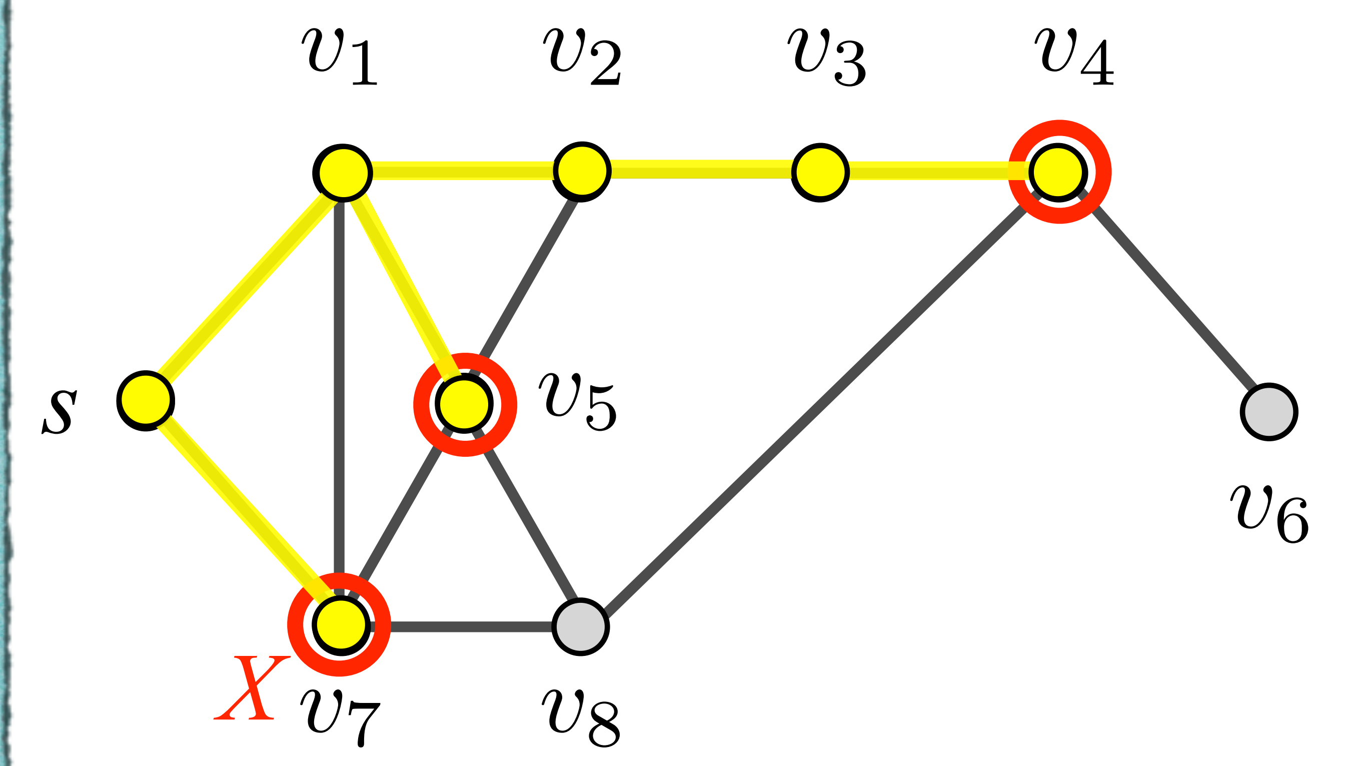
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

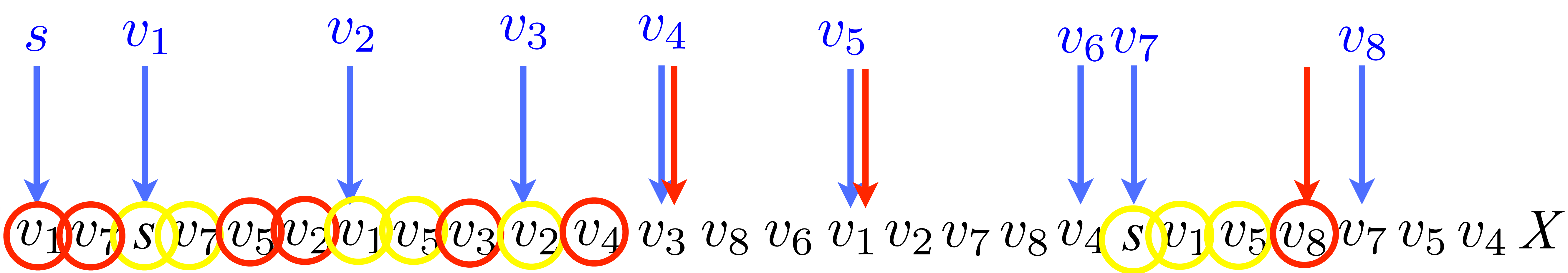
1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

2





Algorithmus 3.7

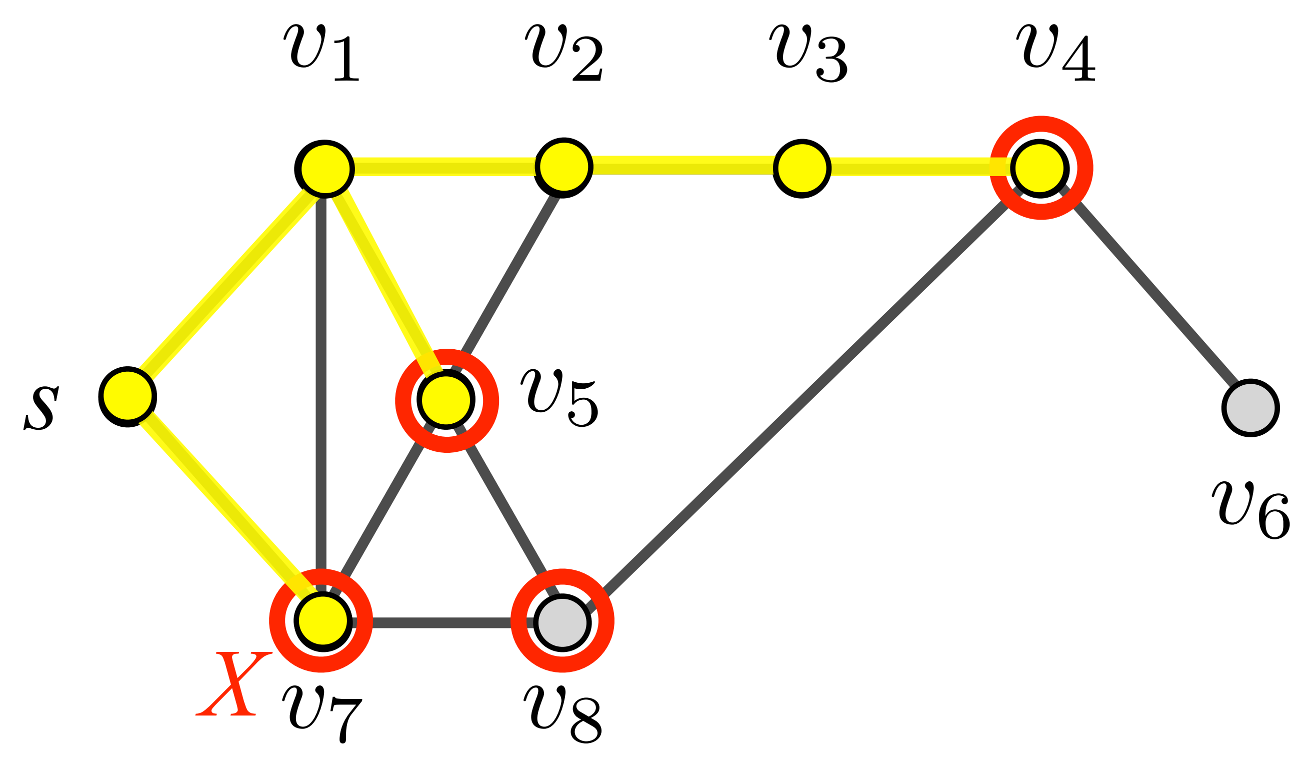
INPUT: Graph G = (V,E), Knoten s

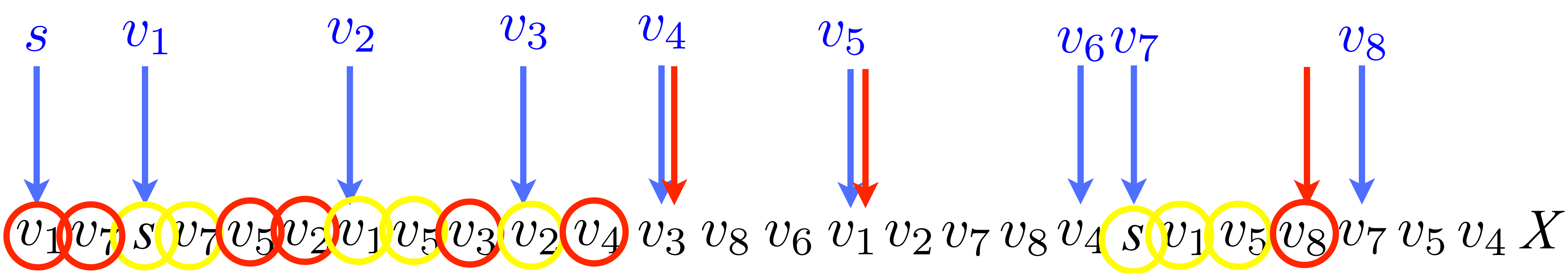
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

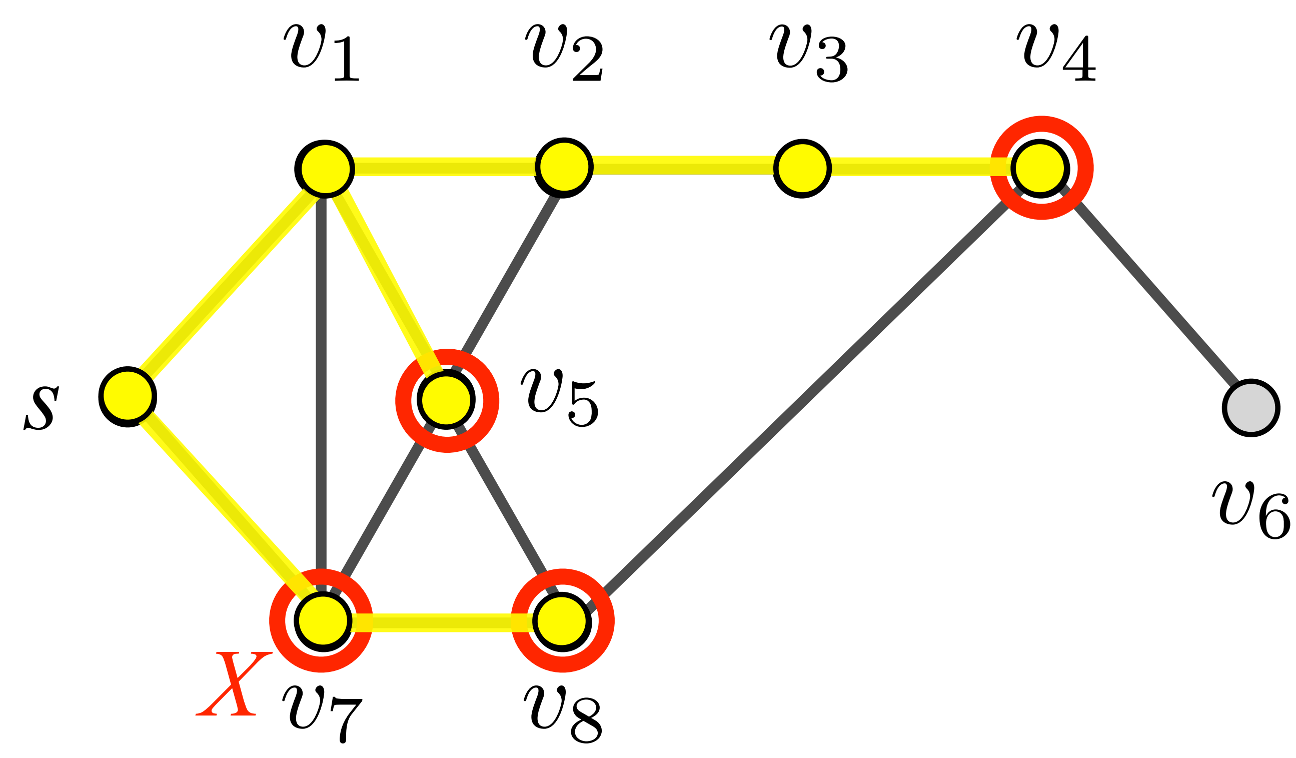
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

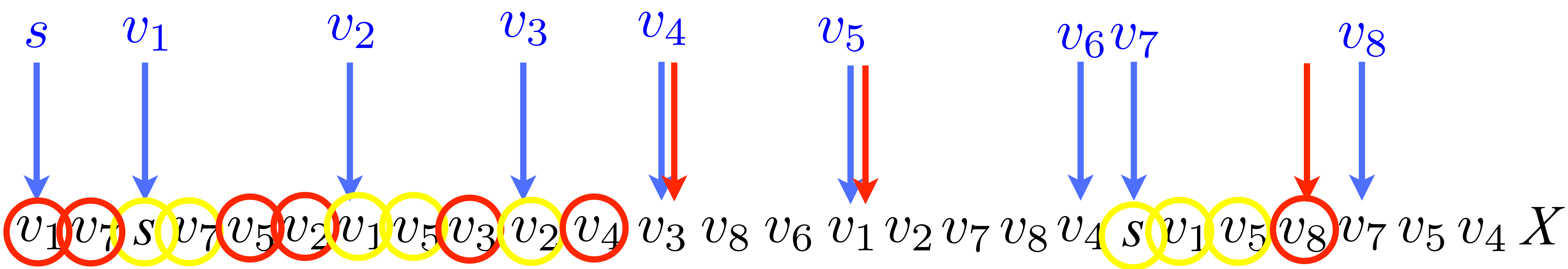
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP

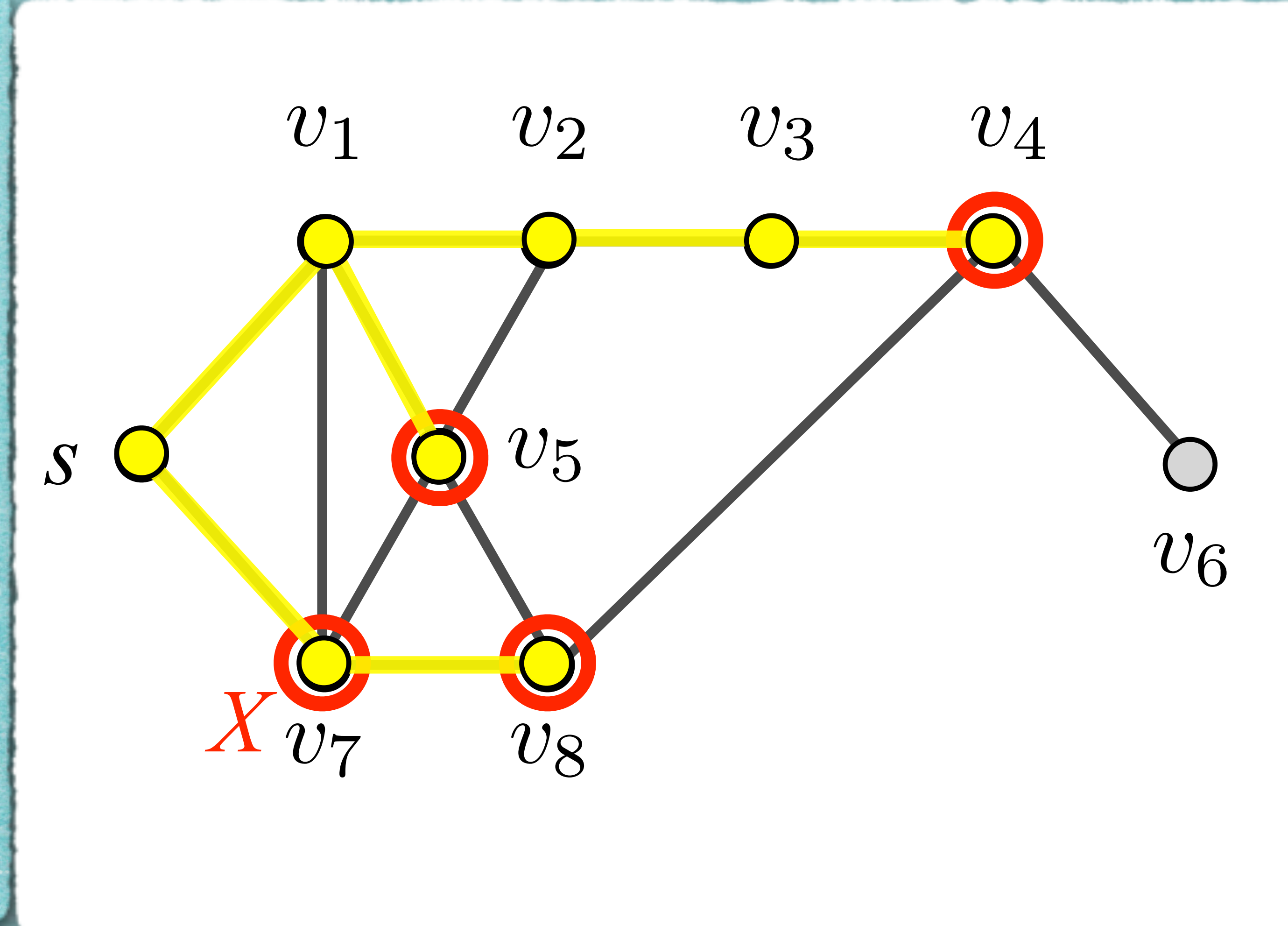


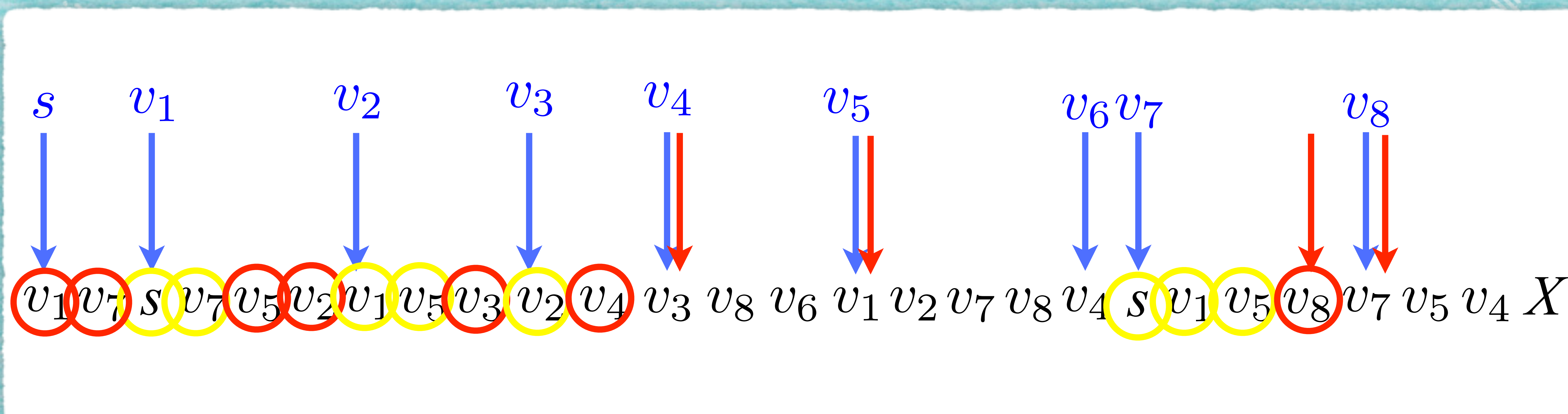


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP

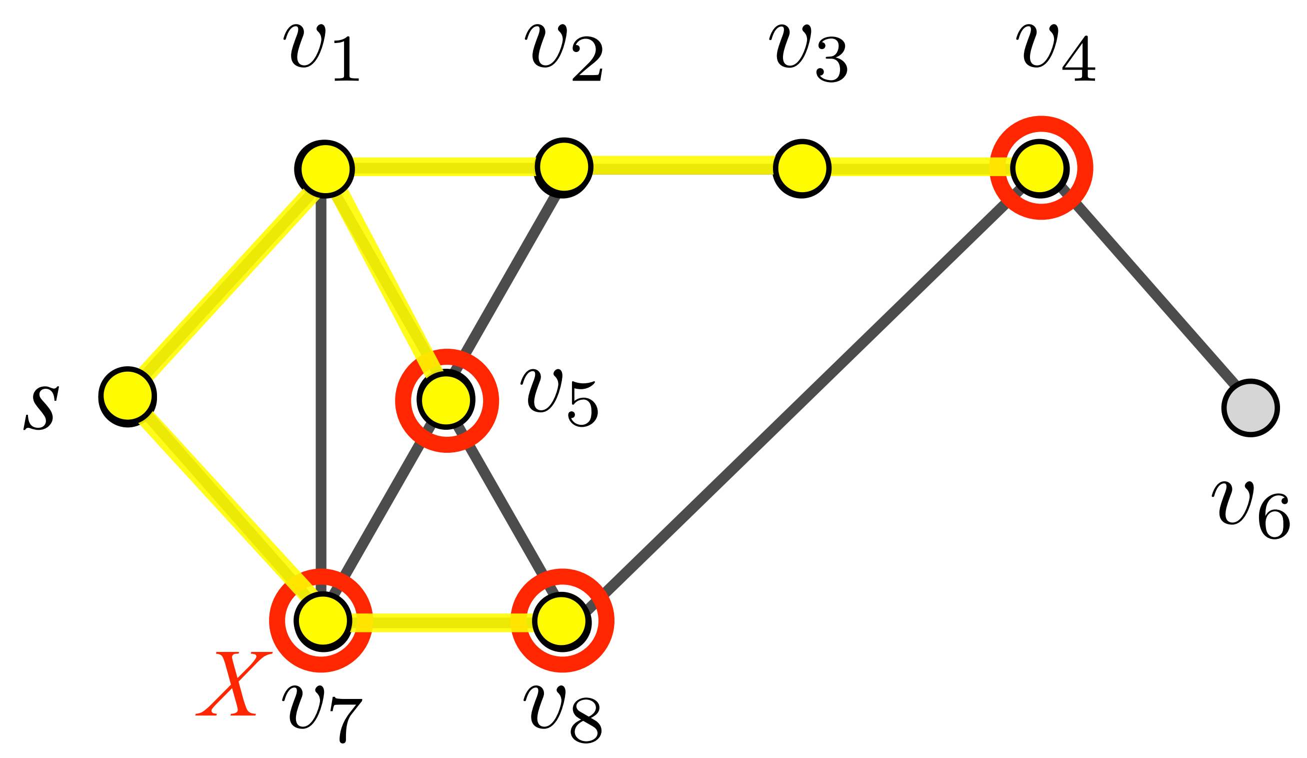


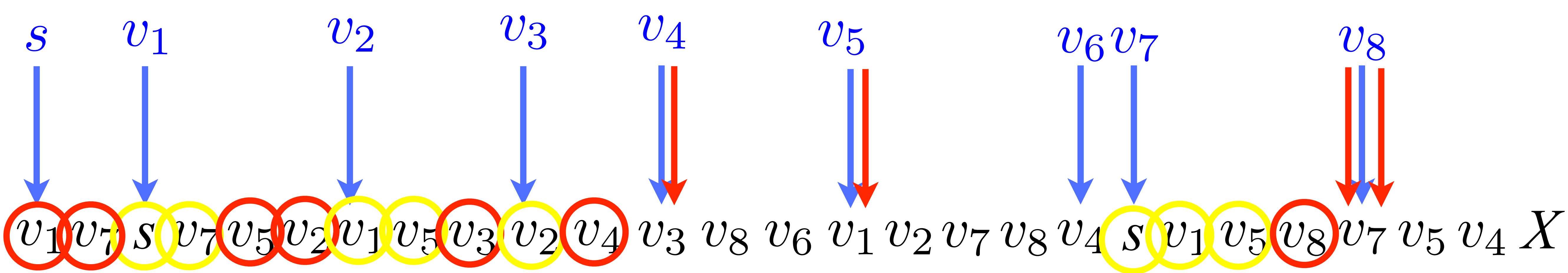


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
   }
3. STOP





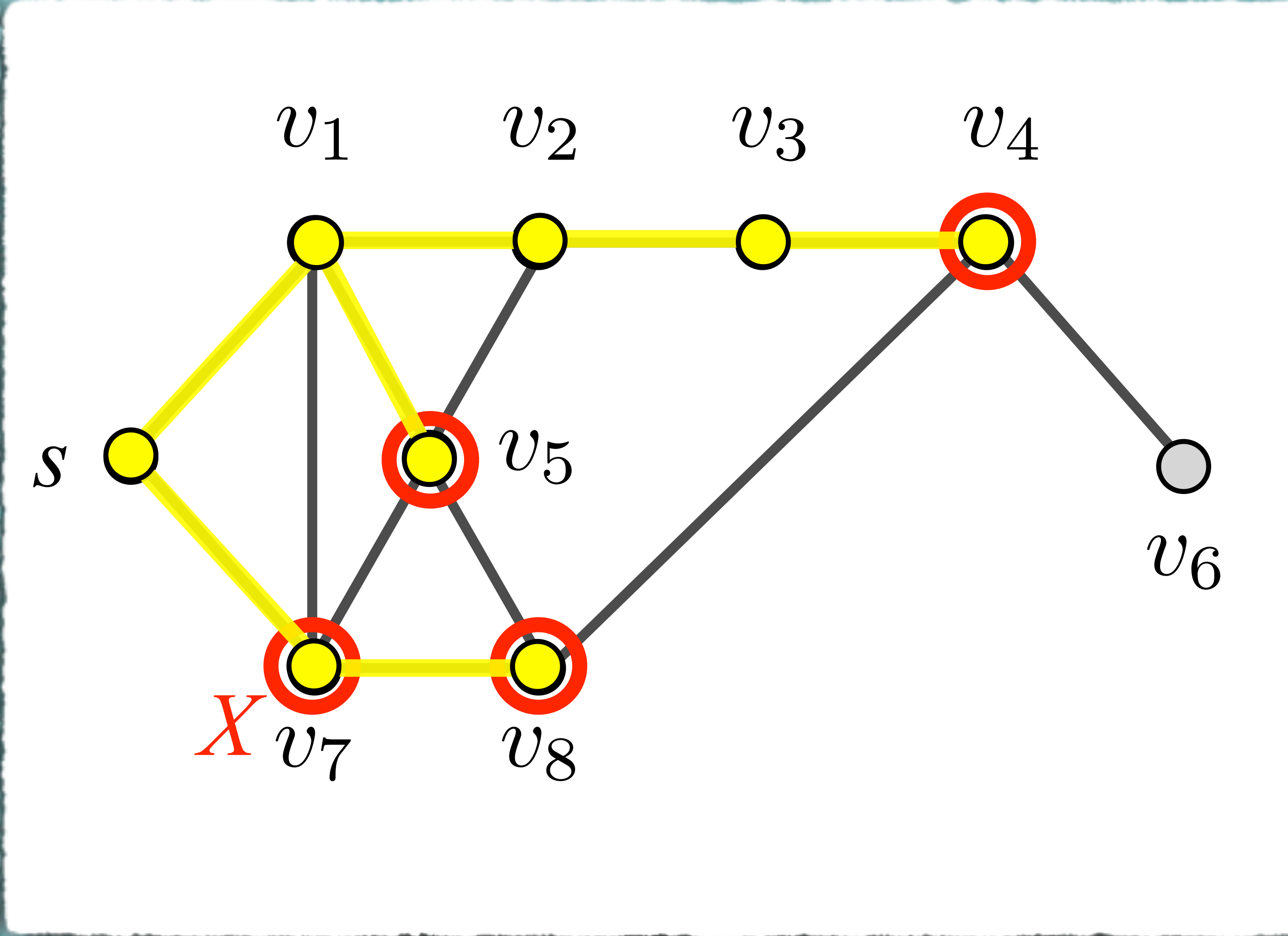
Algorithmus 3.7

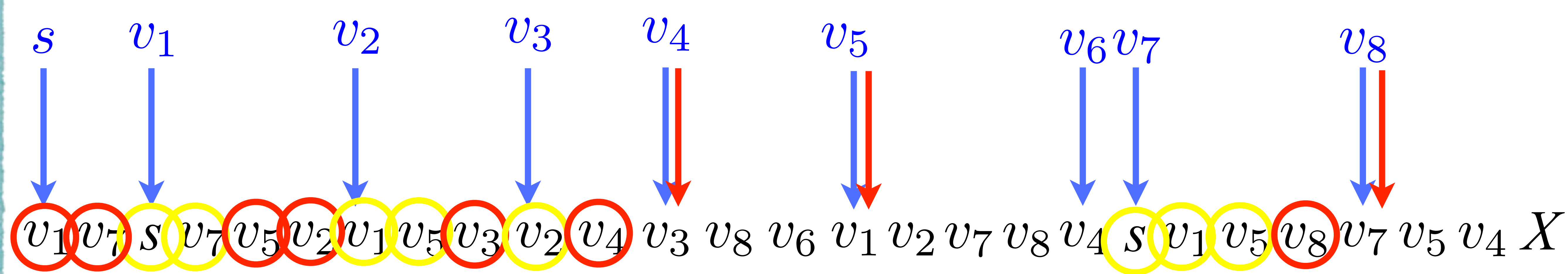
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

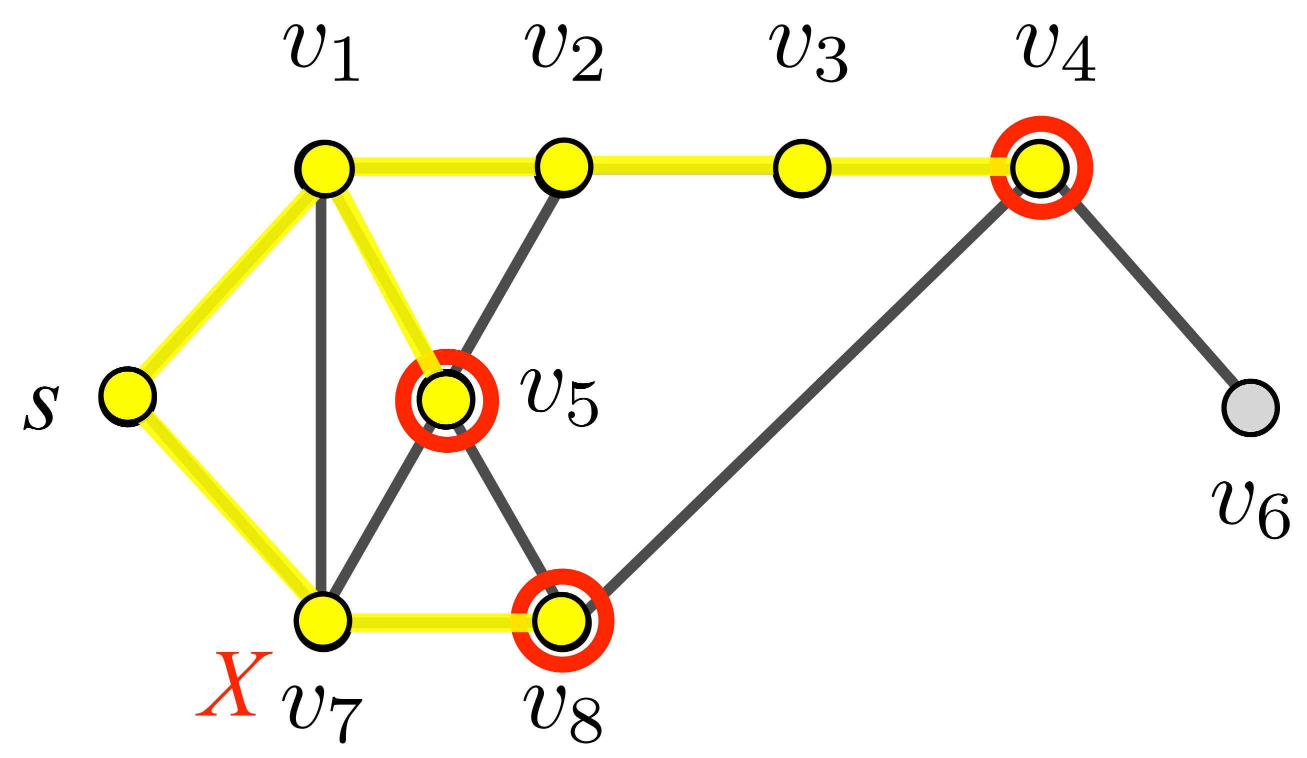
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

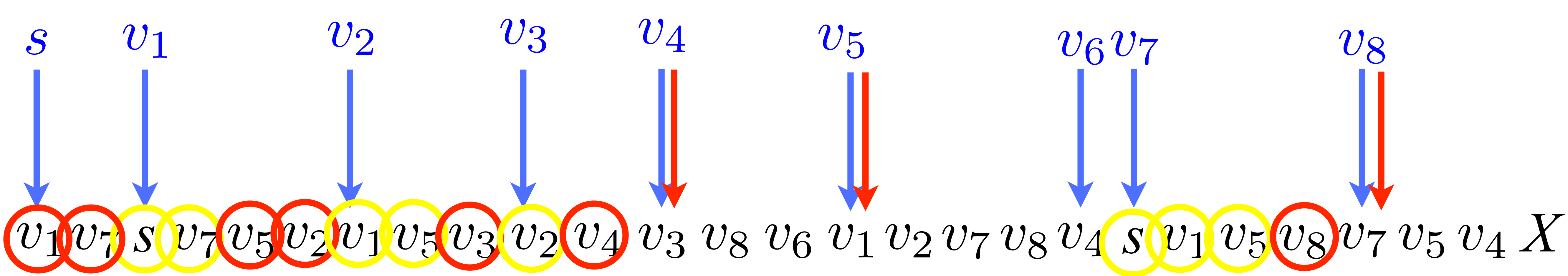
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





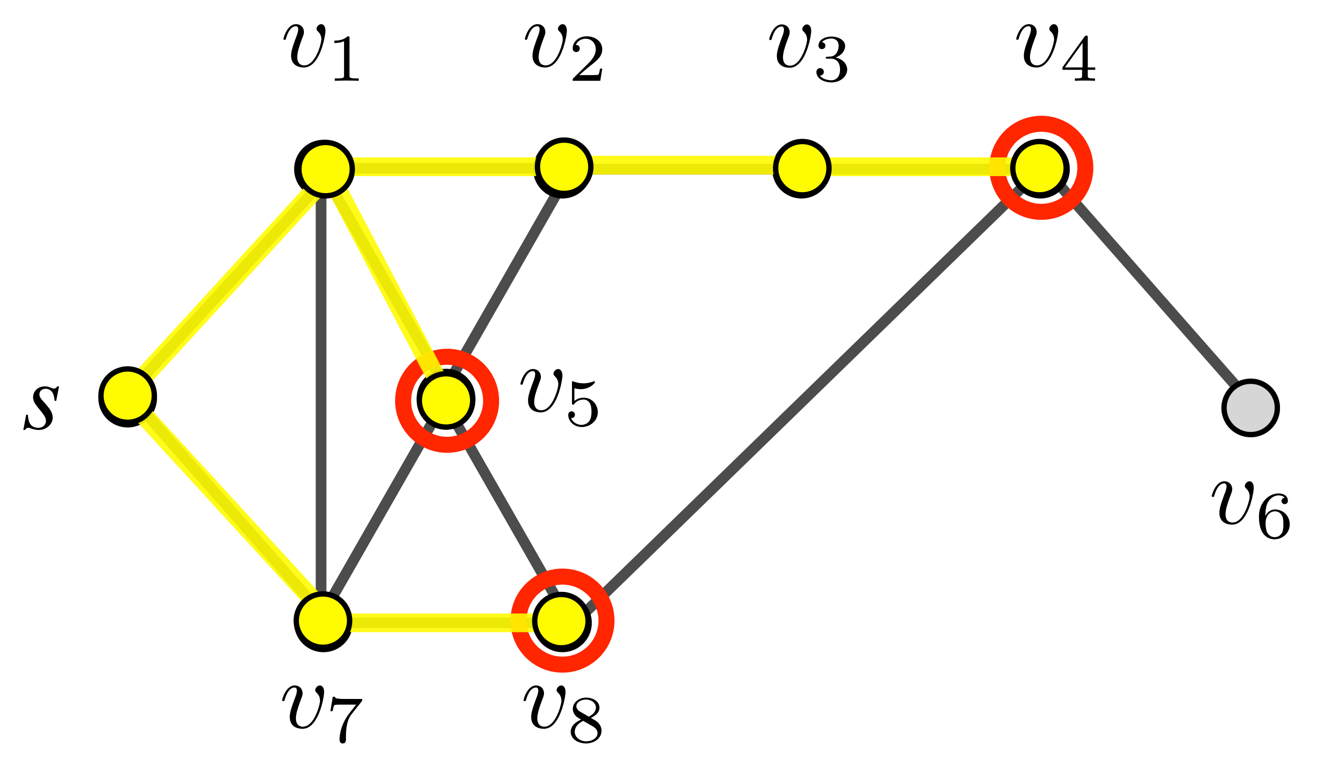
Algorithmus 3.7

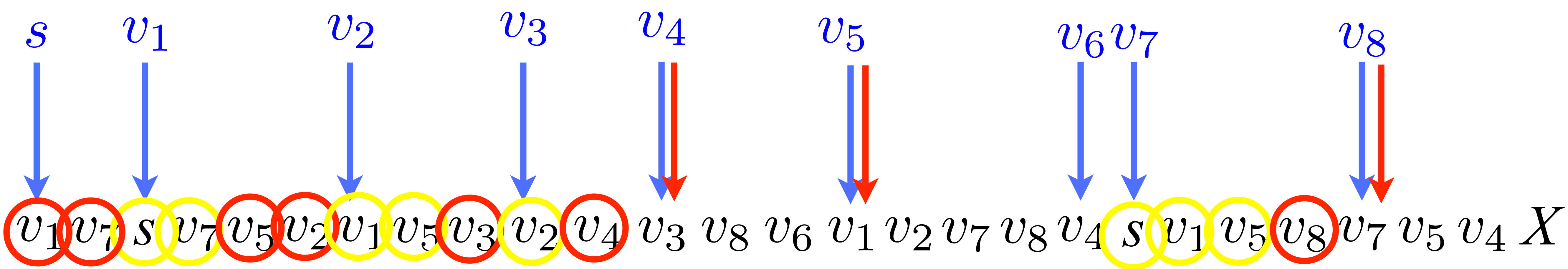
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v ∈ R
   - 2.2. IF (es gibt kein w ∈ V\Y mit e={v,w} ∈ E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

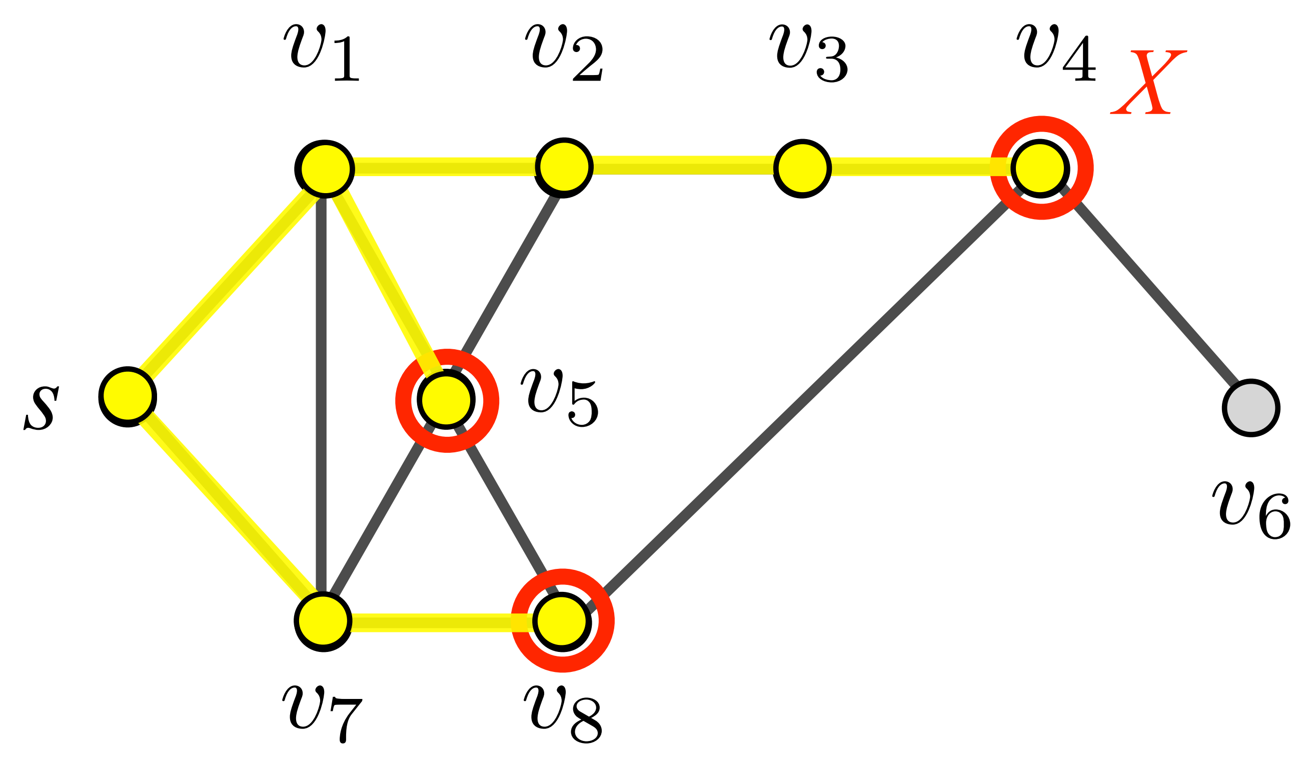


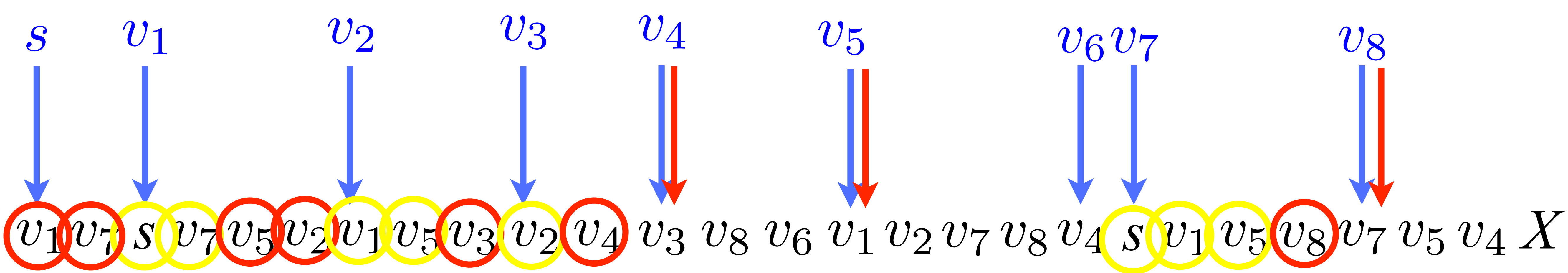


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

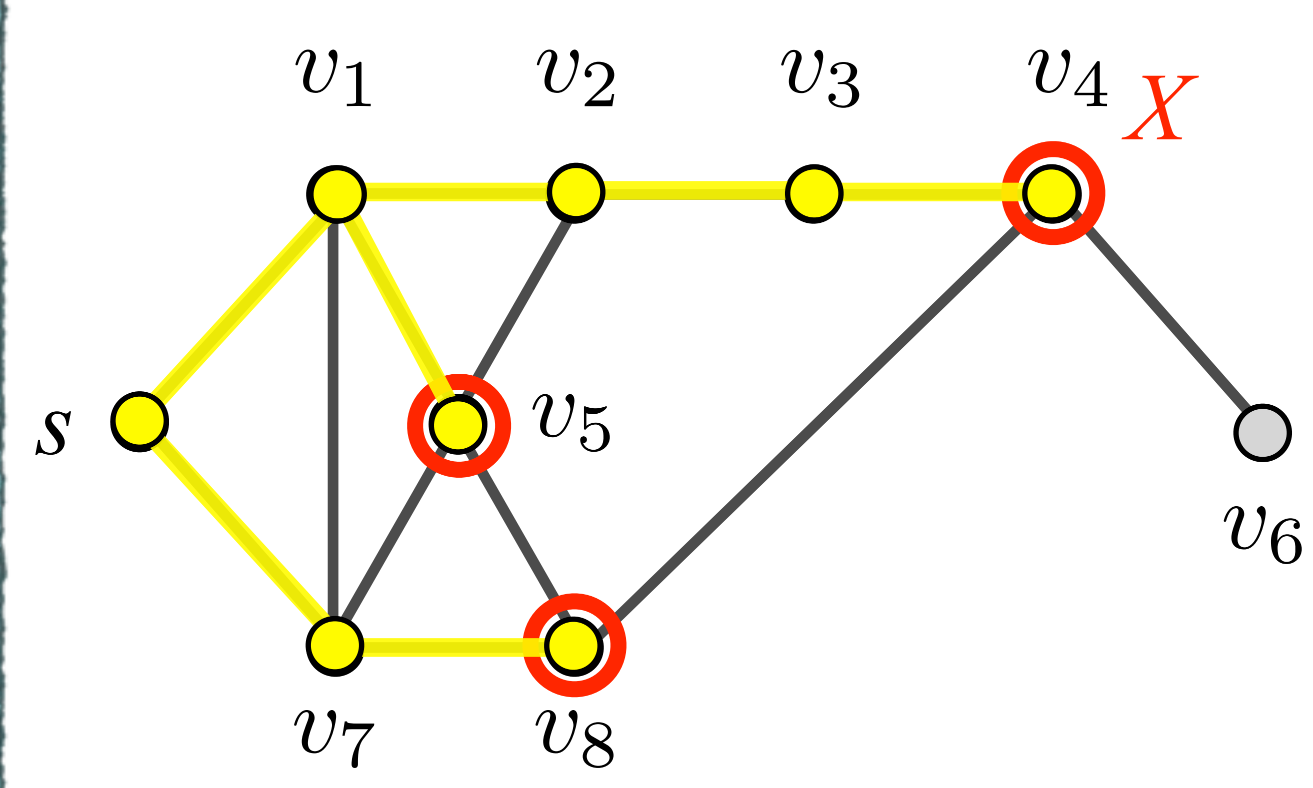
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

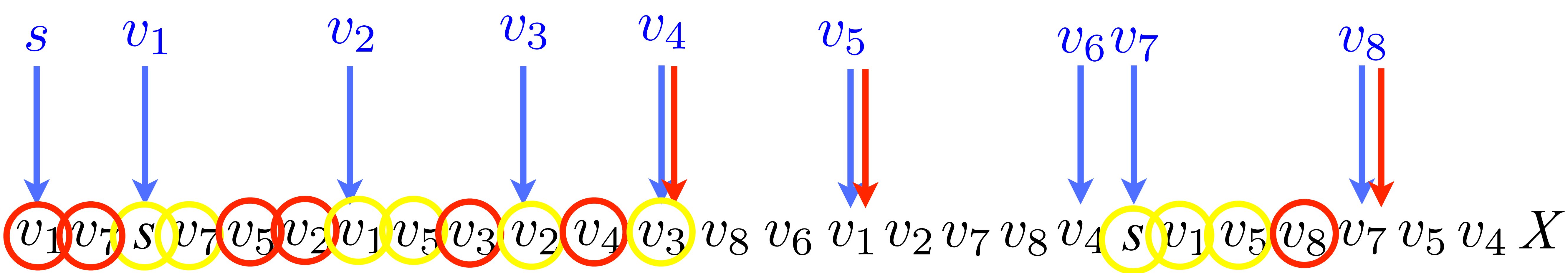
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

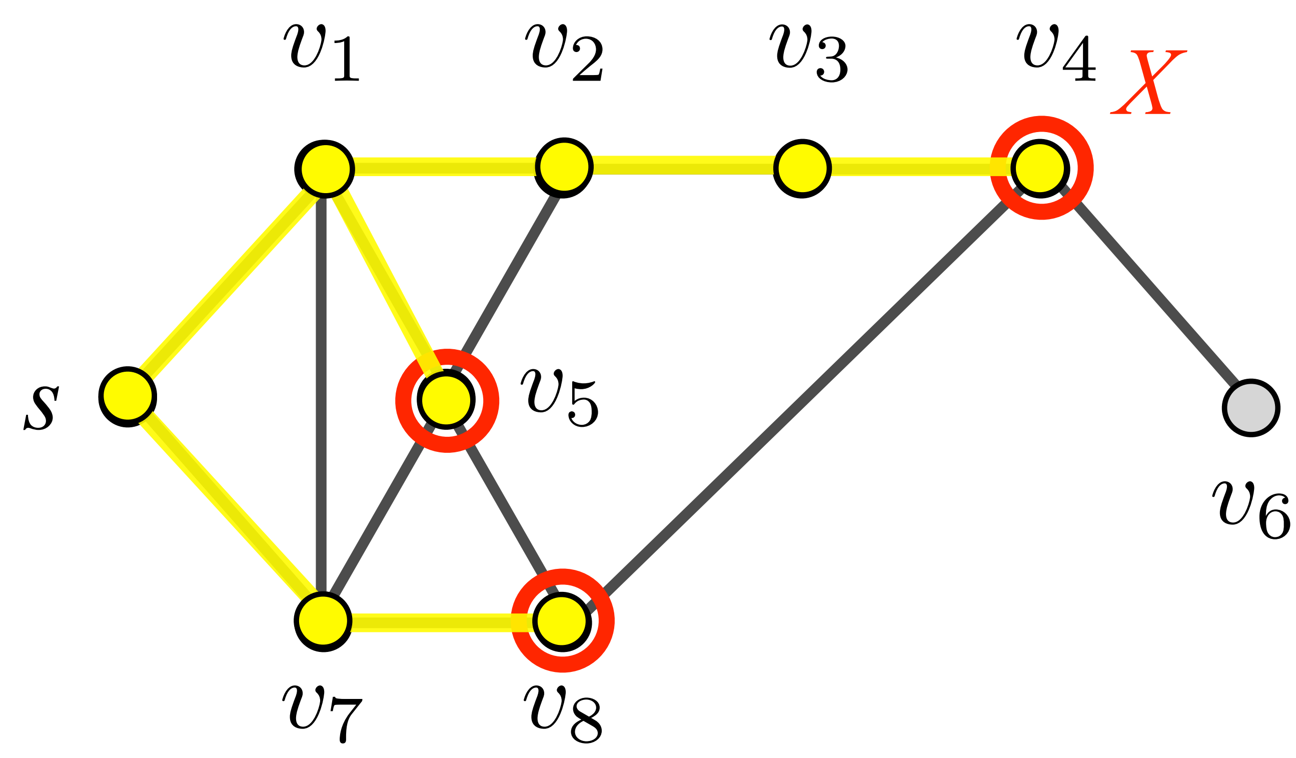
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

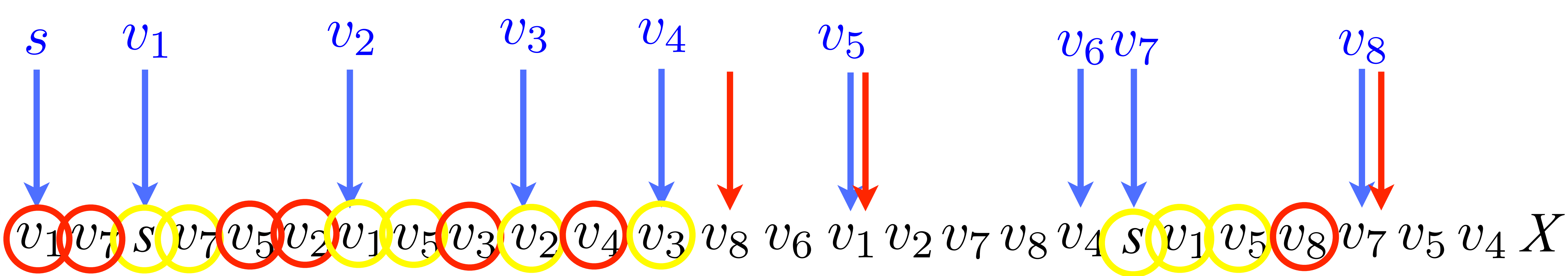
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

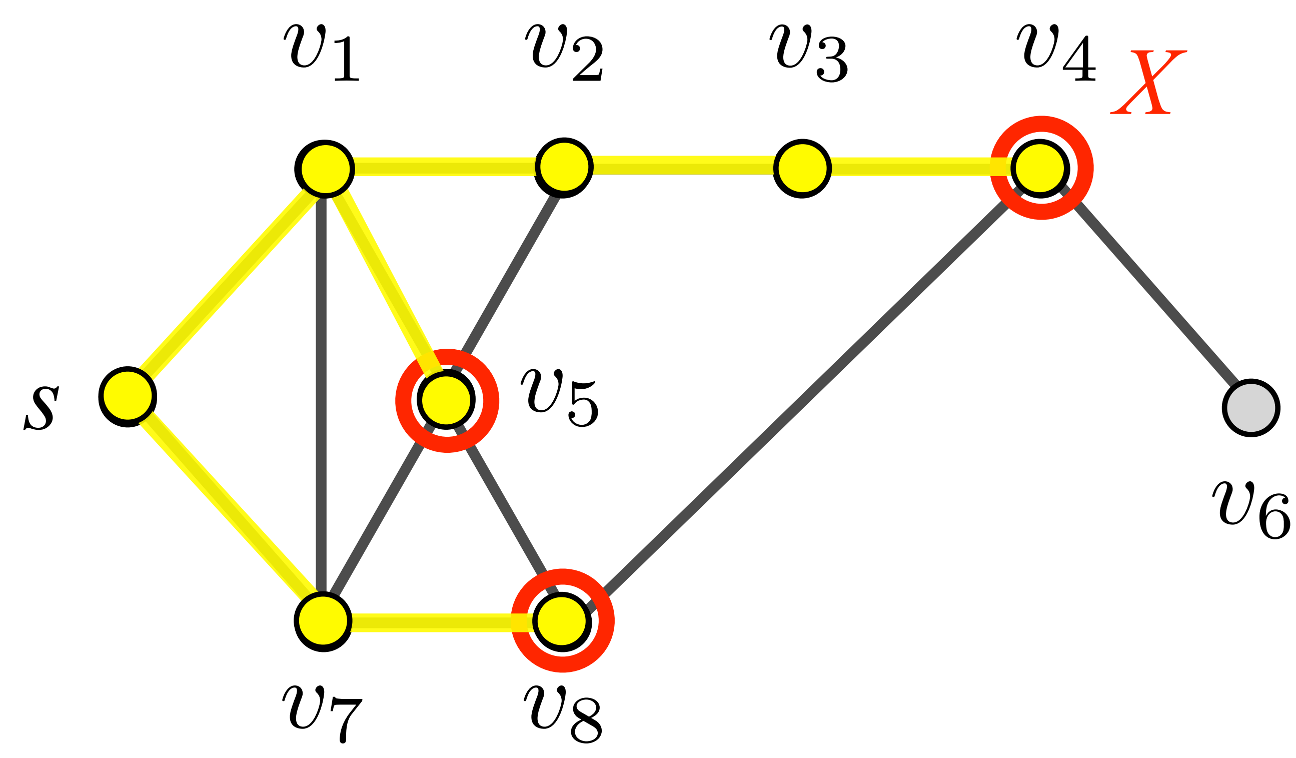


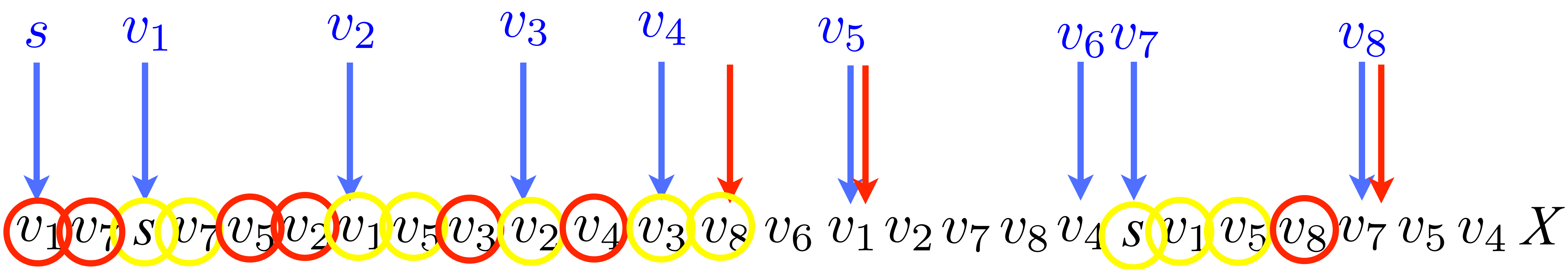


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

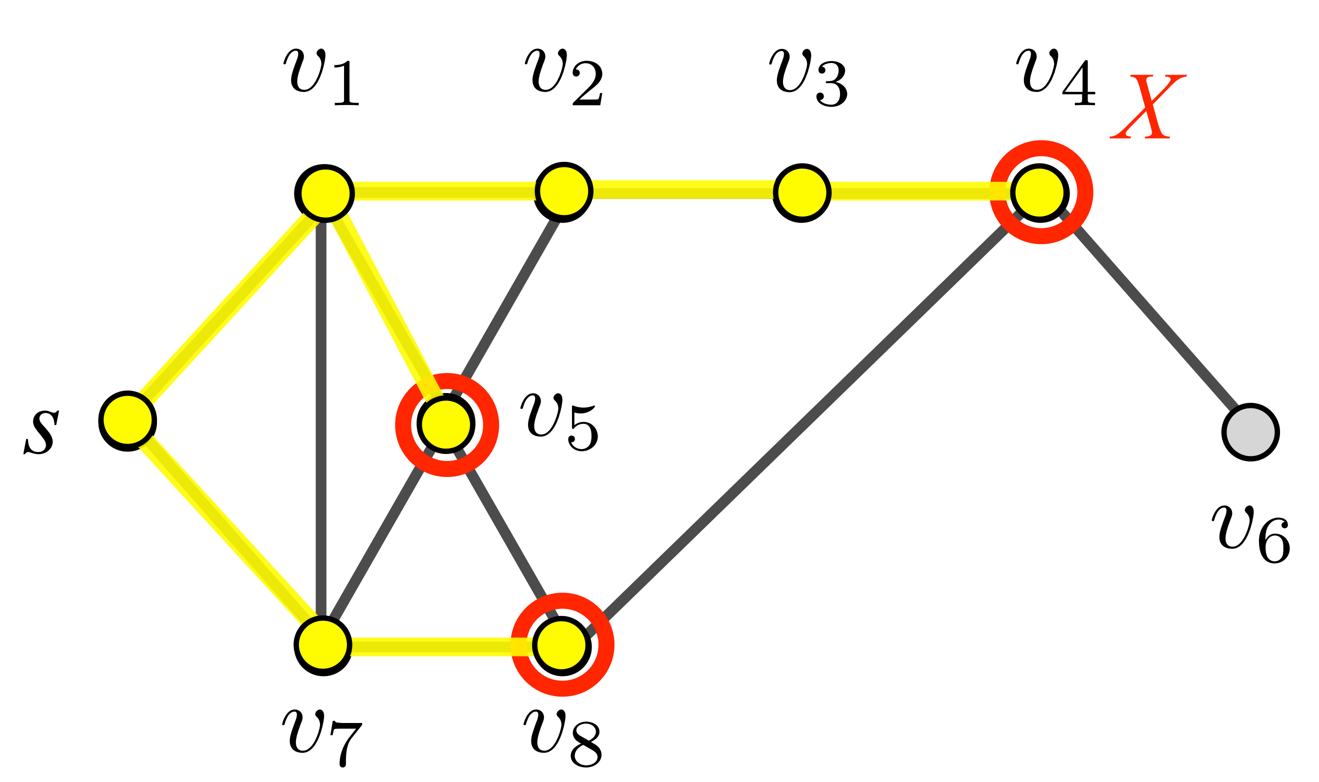
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

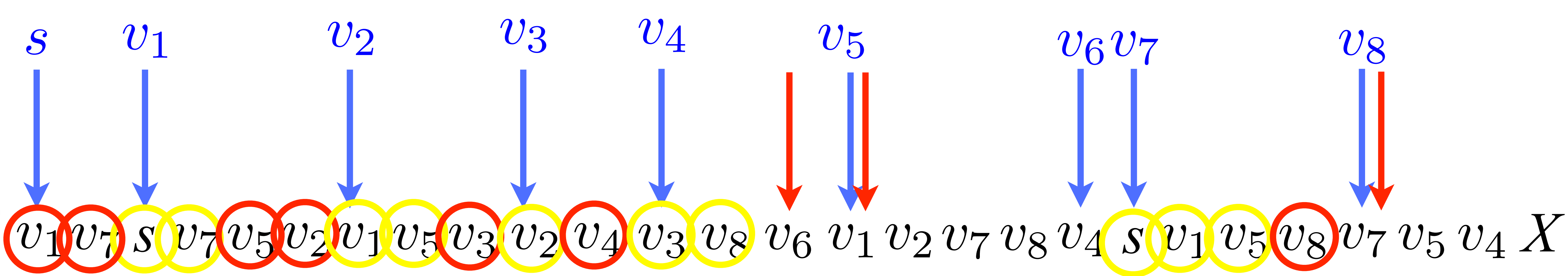
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP

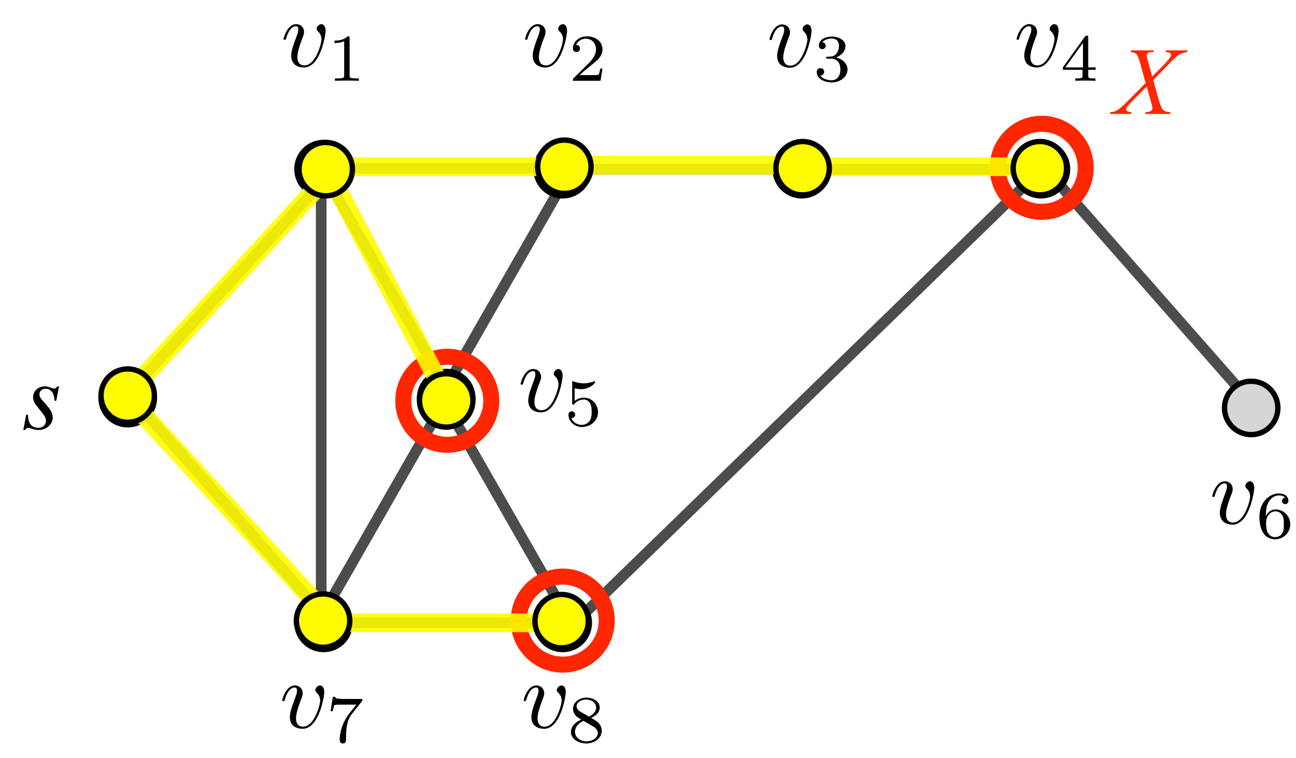


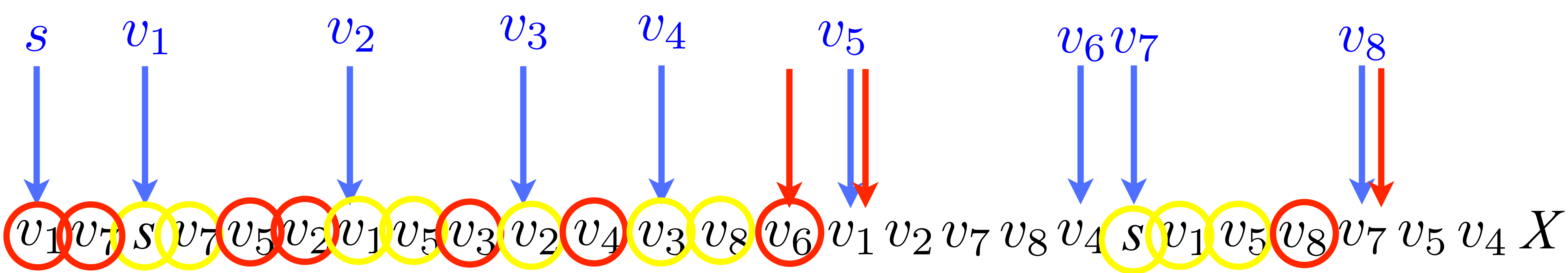


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈ R
   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
      2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
      }
   }
3. STOP

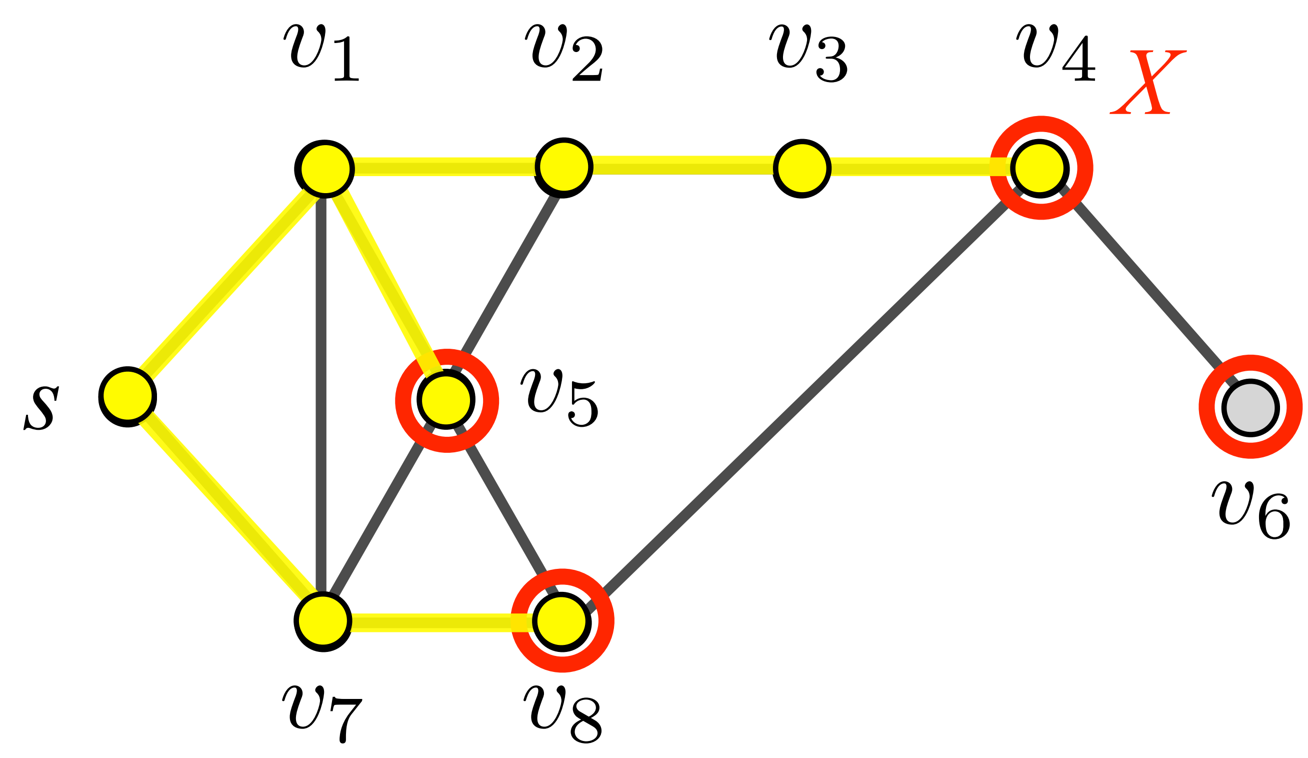


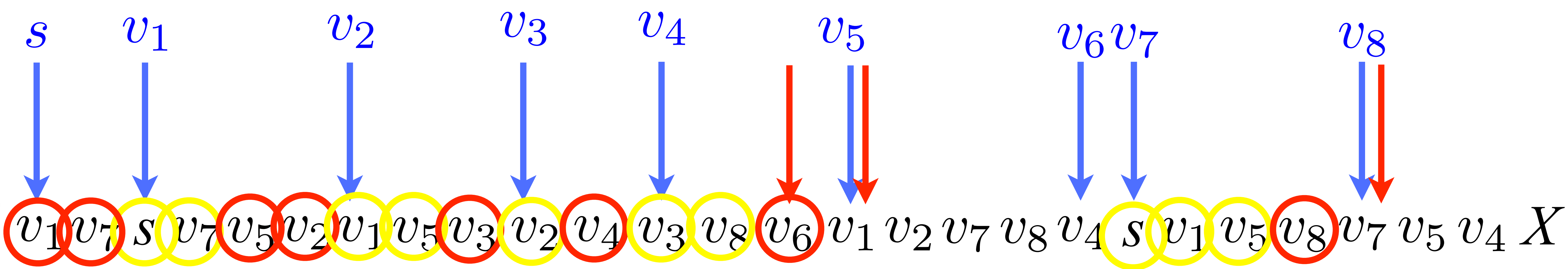


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

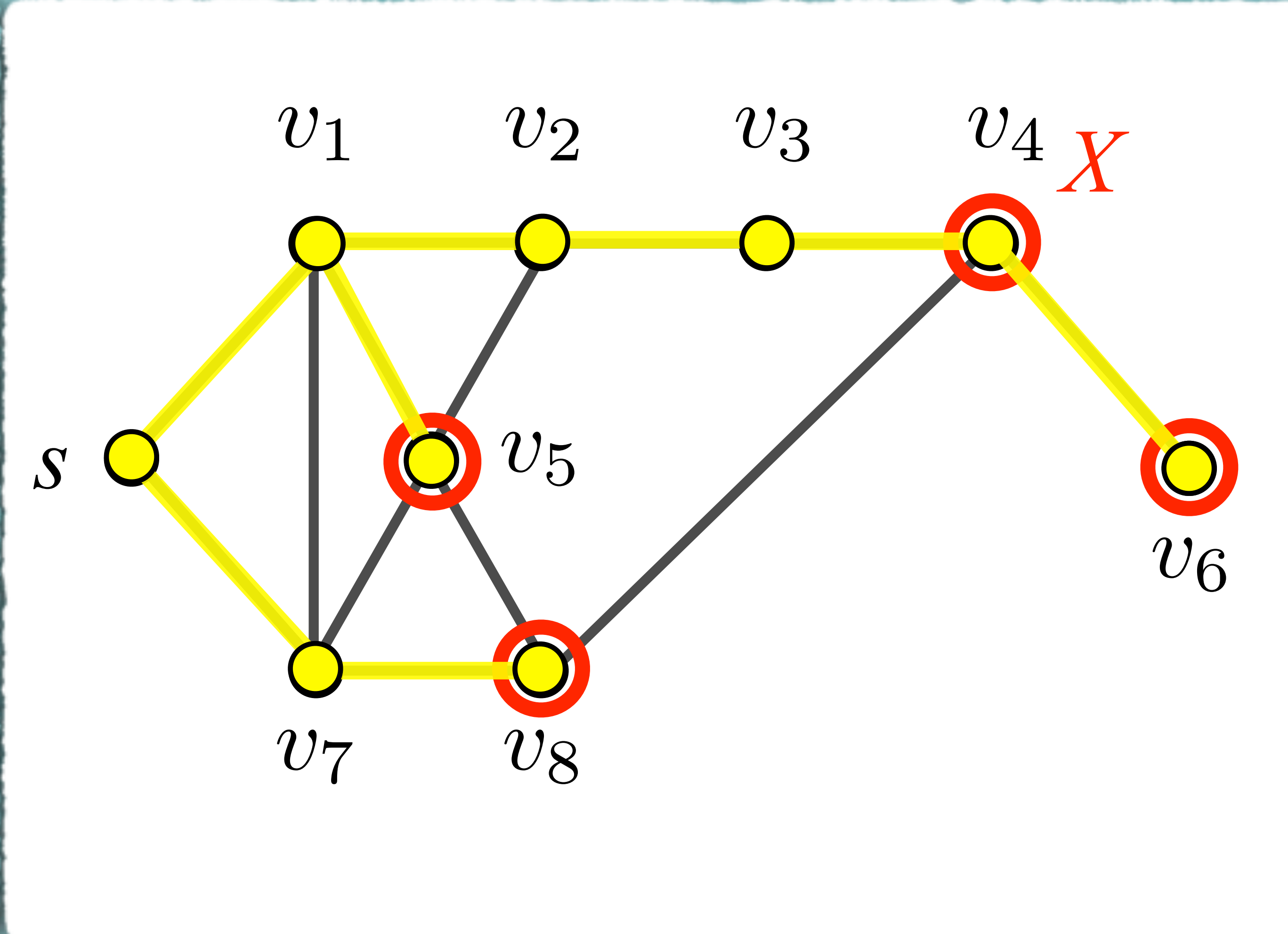
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

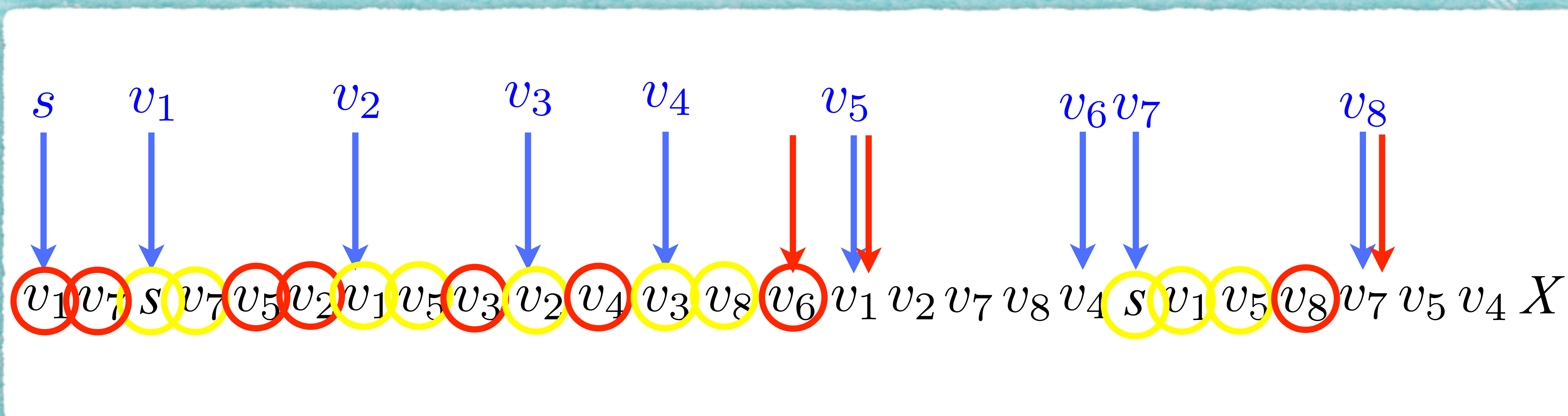
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

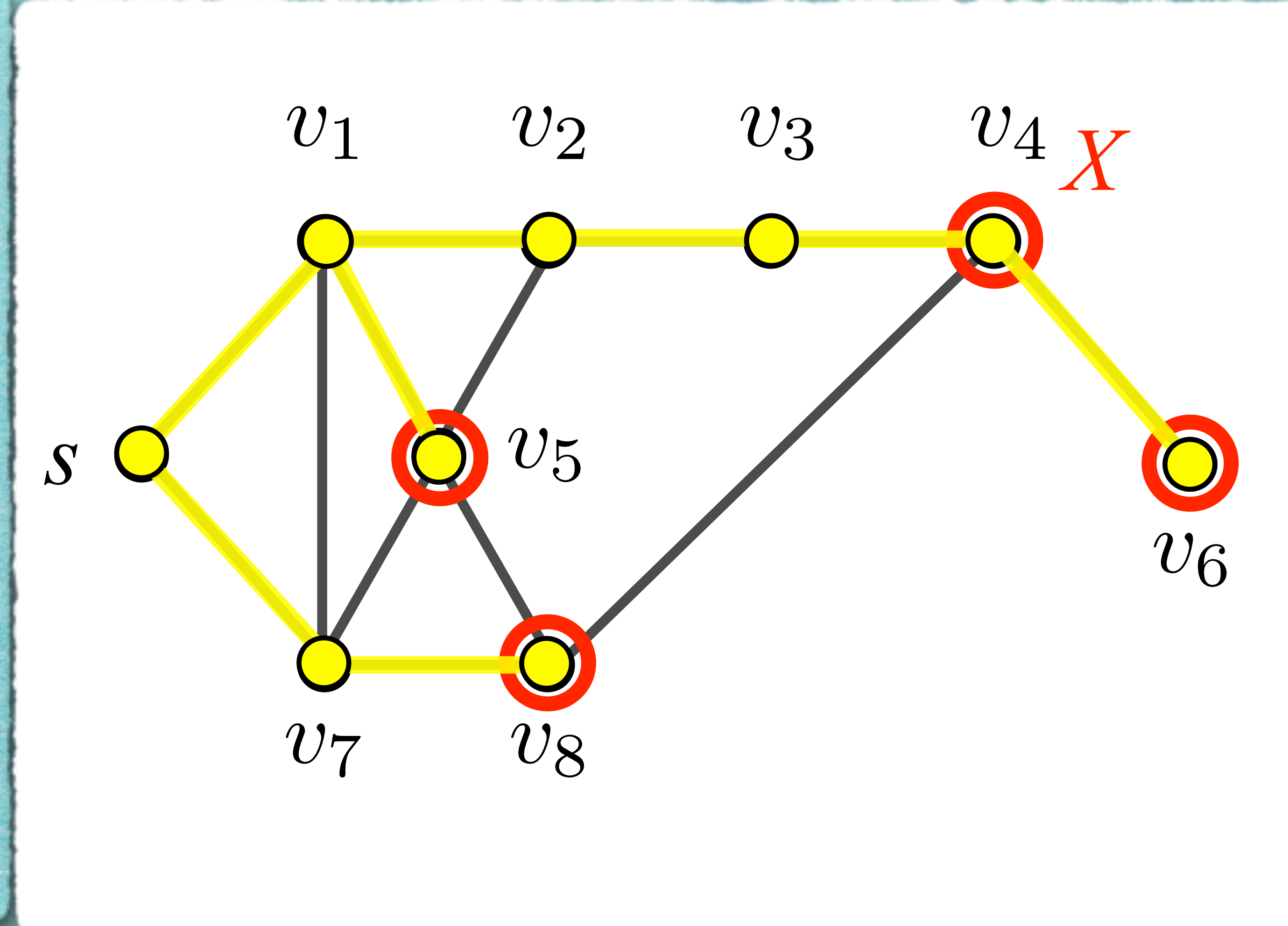
INPUT: Graph G = (V,E), Knoten s

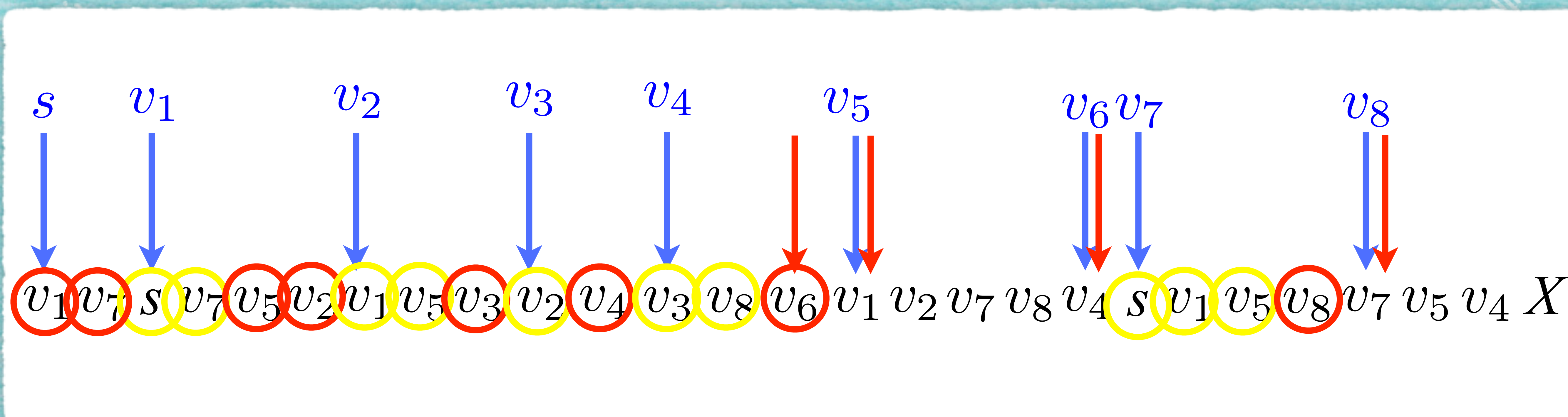
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





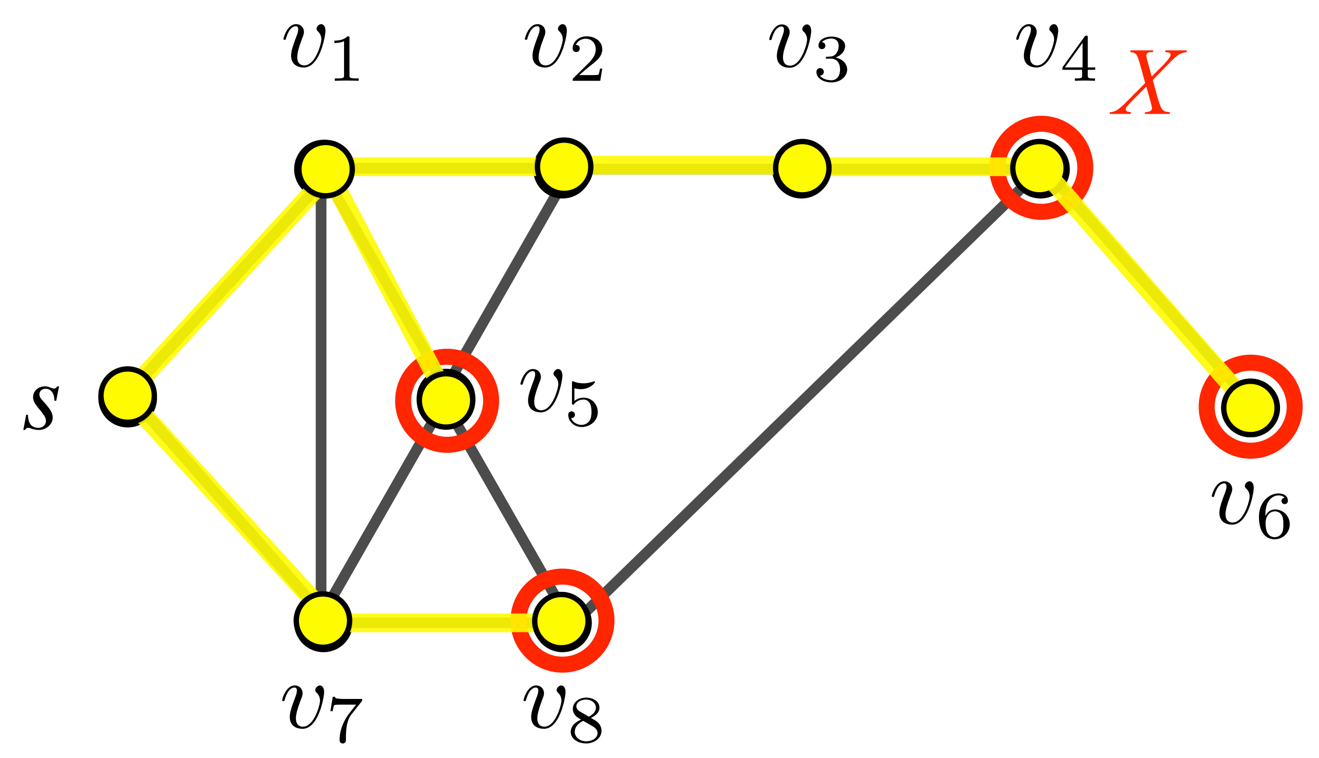
Algorithmus 3.7

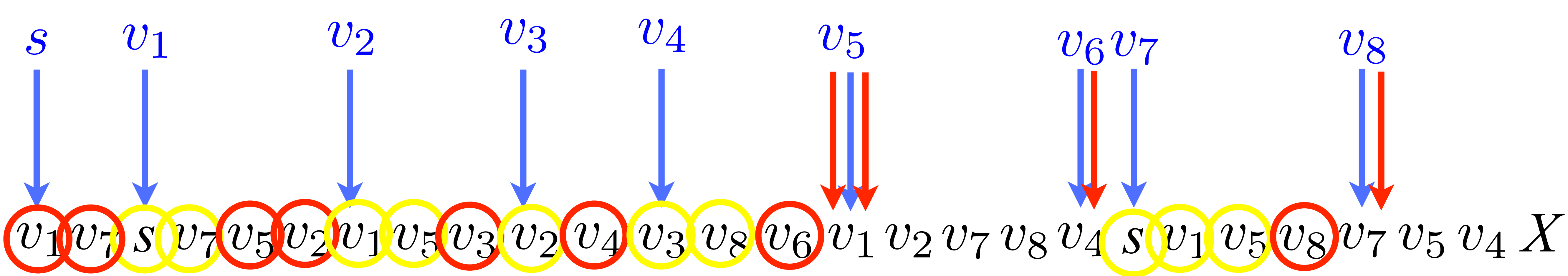
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

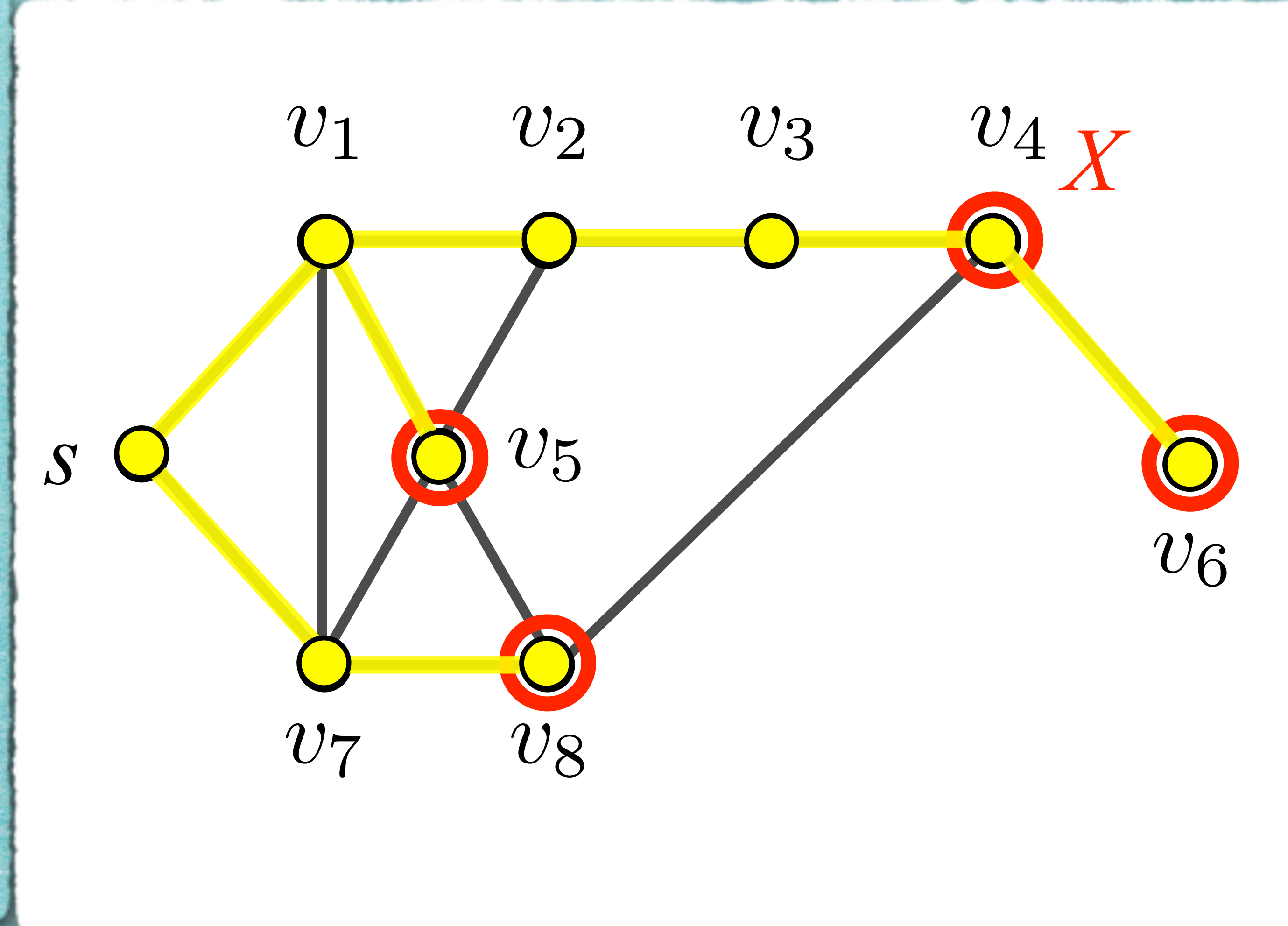
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

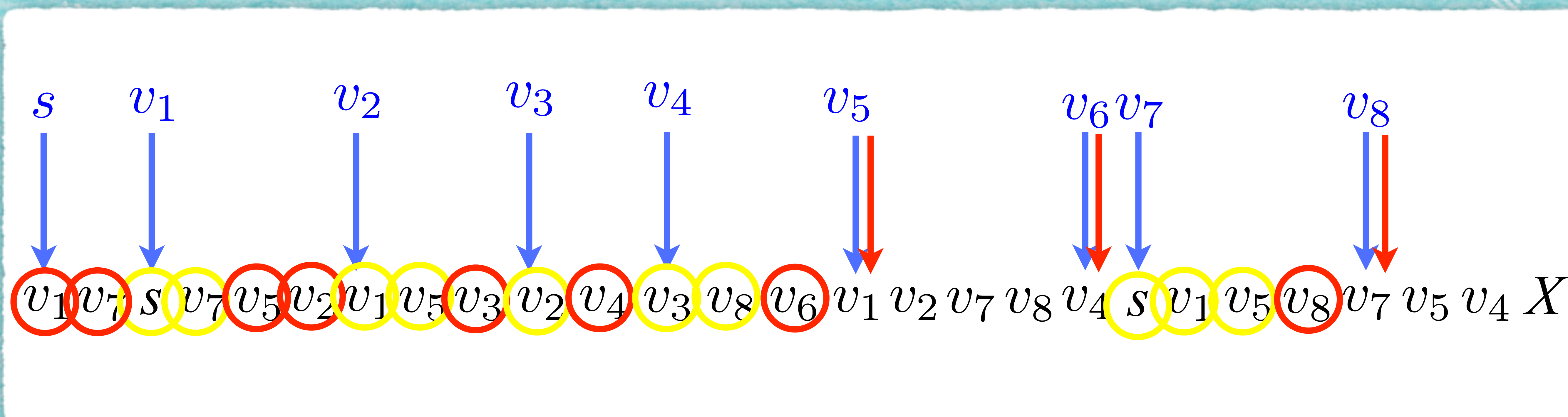
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V\backslash Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V\backslash Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP



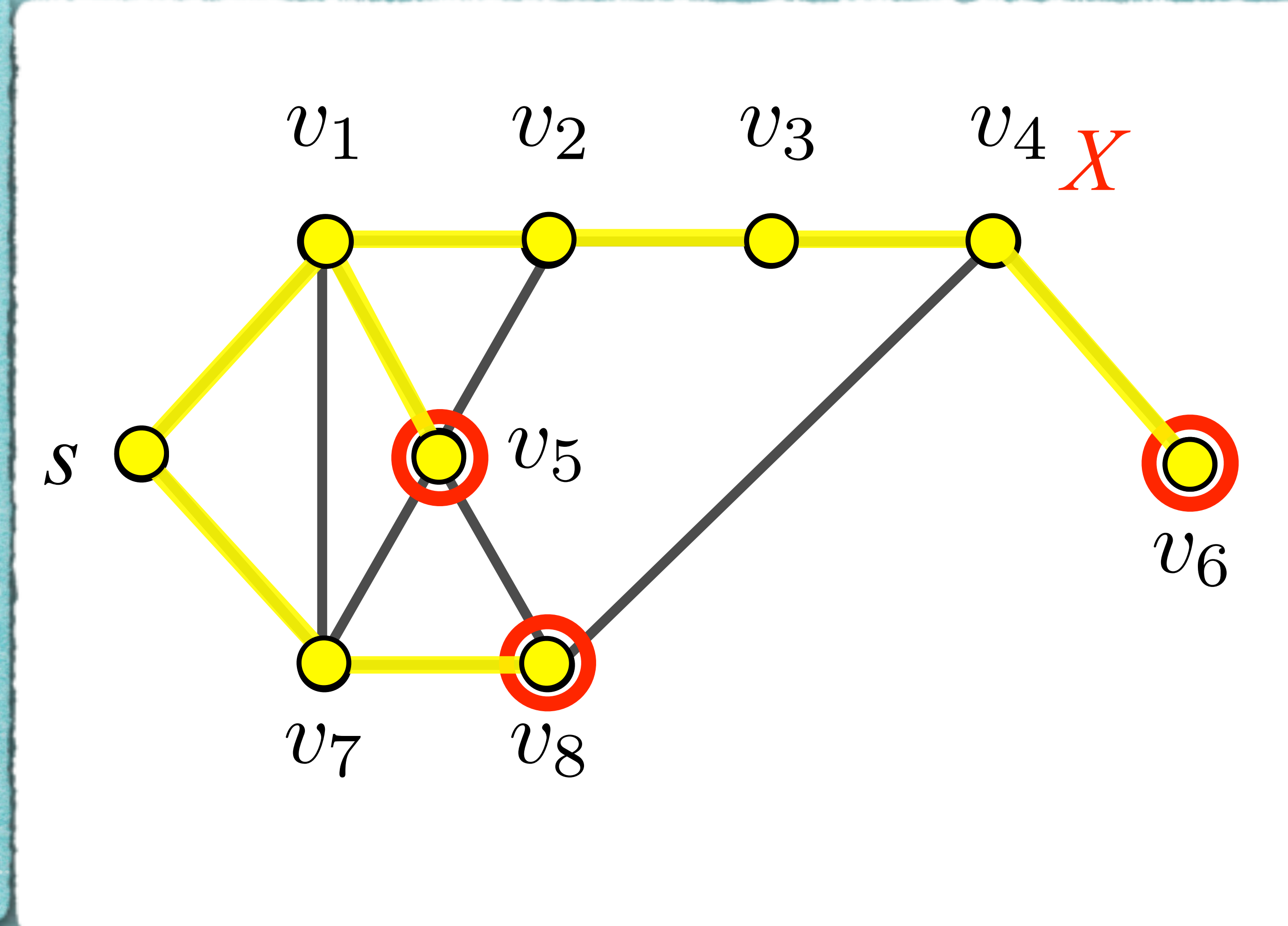


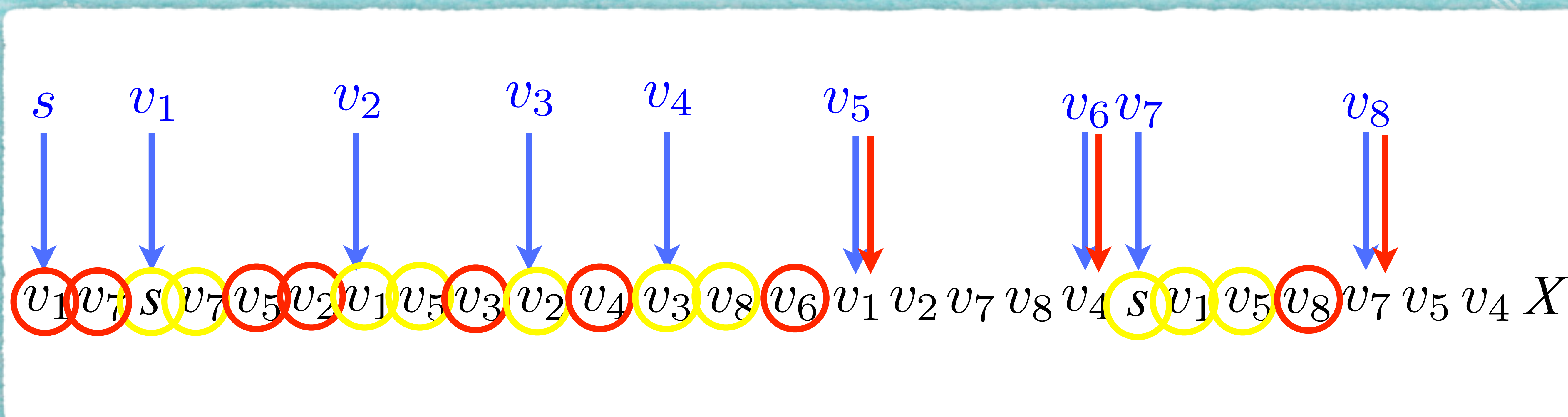
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

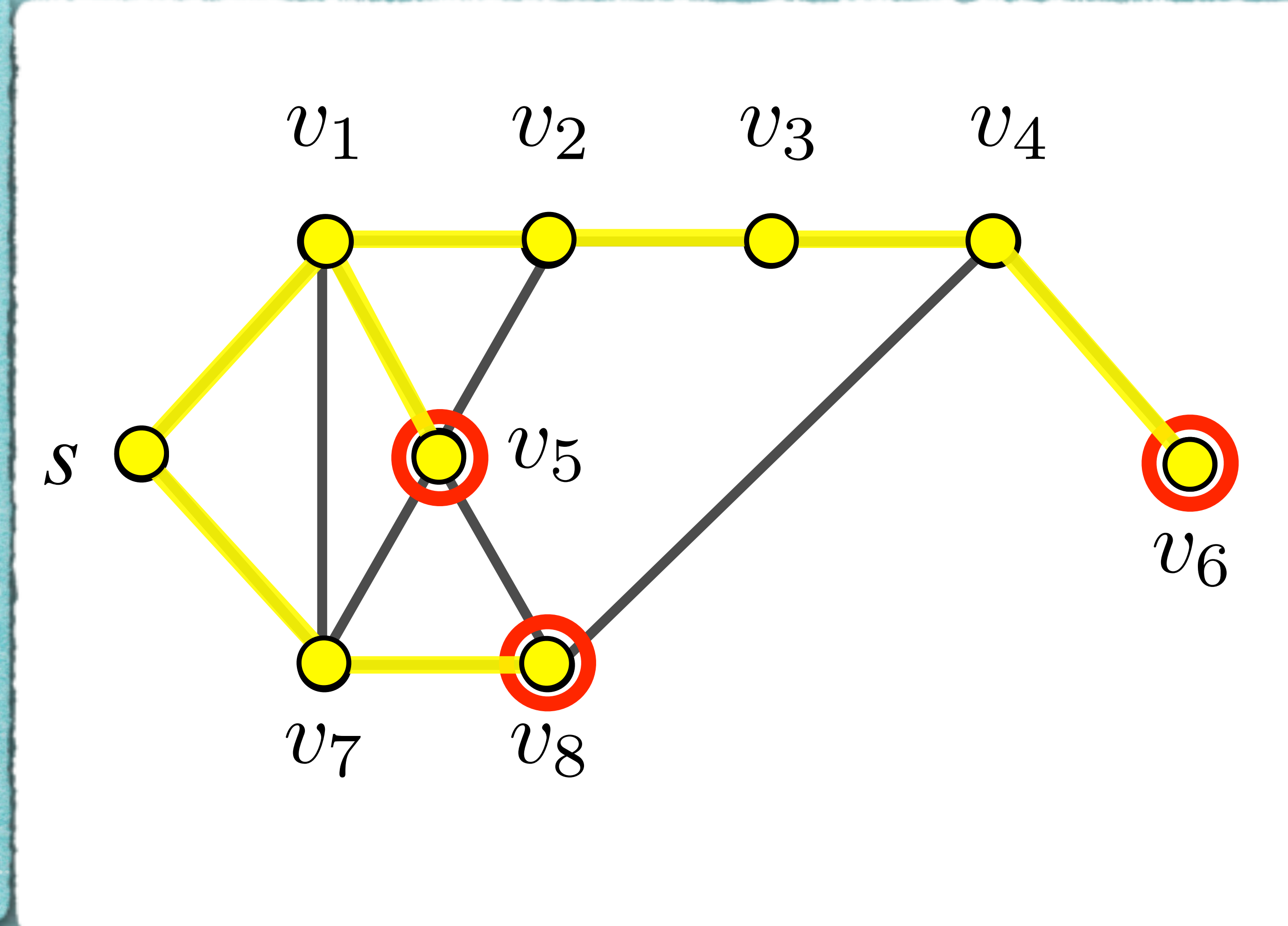
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

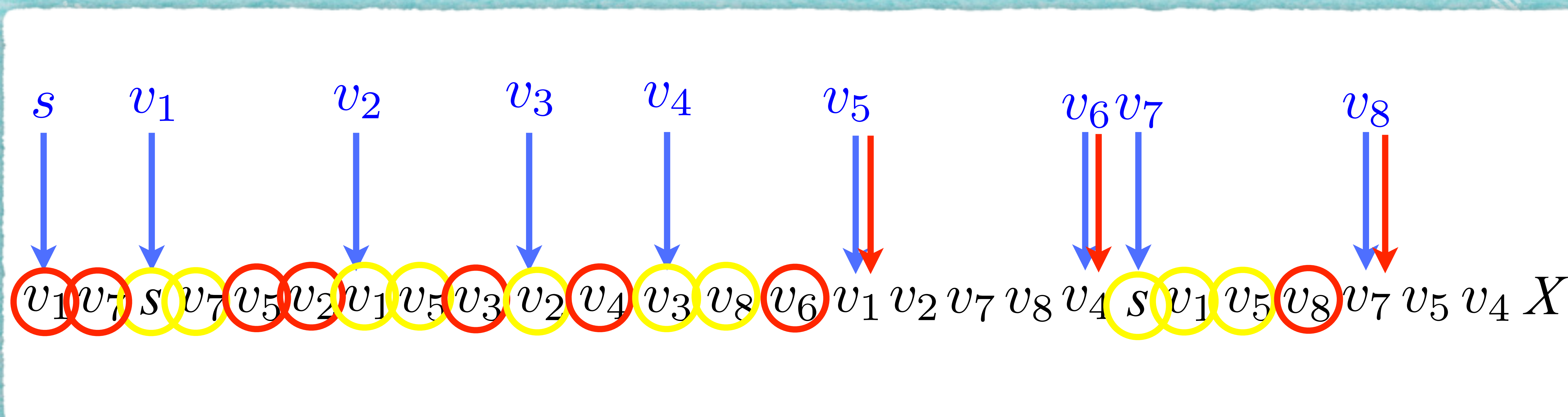
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





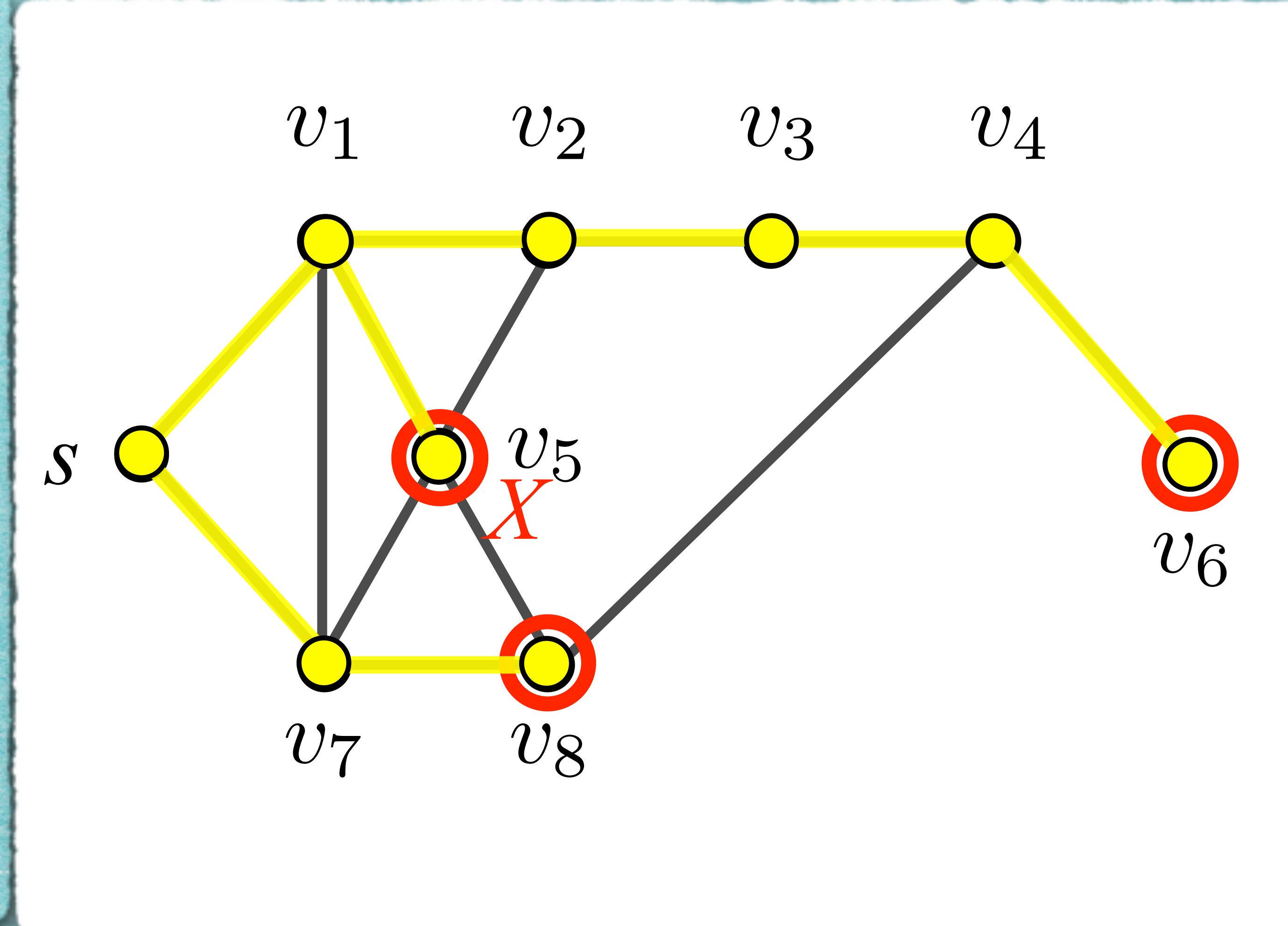
Algorithmus 3.7

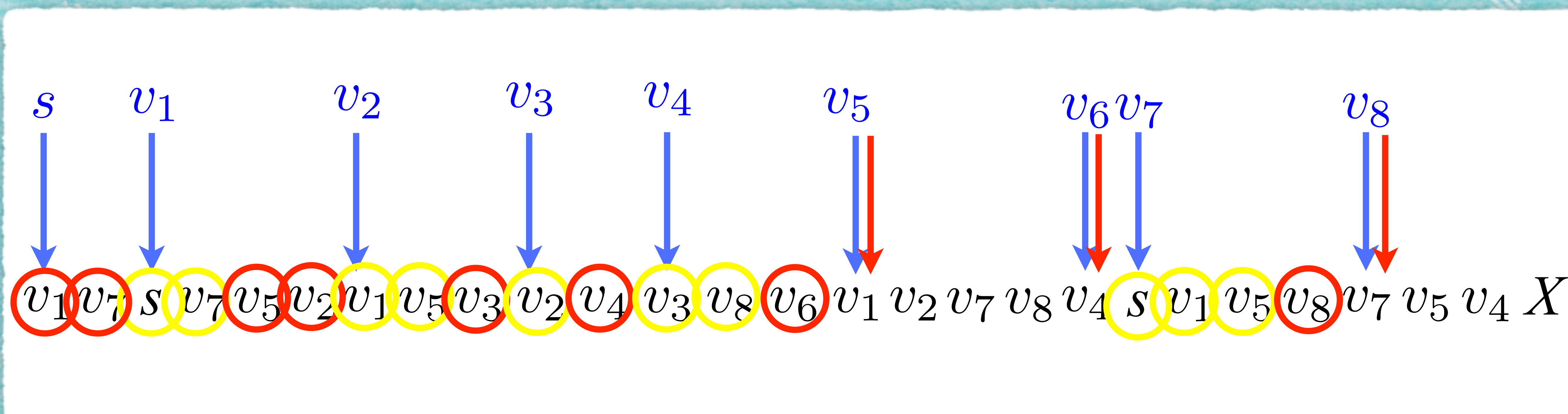
INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R\neq\emptyset$) DO {
    - 2.1. Wähle $v\in R$
    - 2.2. IF (es gibt kein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$) THEN
        - 2.2.1. $R:=R\setminus\{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w\in V\setminus Y$ mit $e=\{v,w\}\in E$
        - 2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$

        }

    }
3. STOP





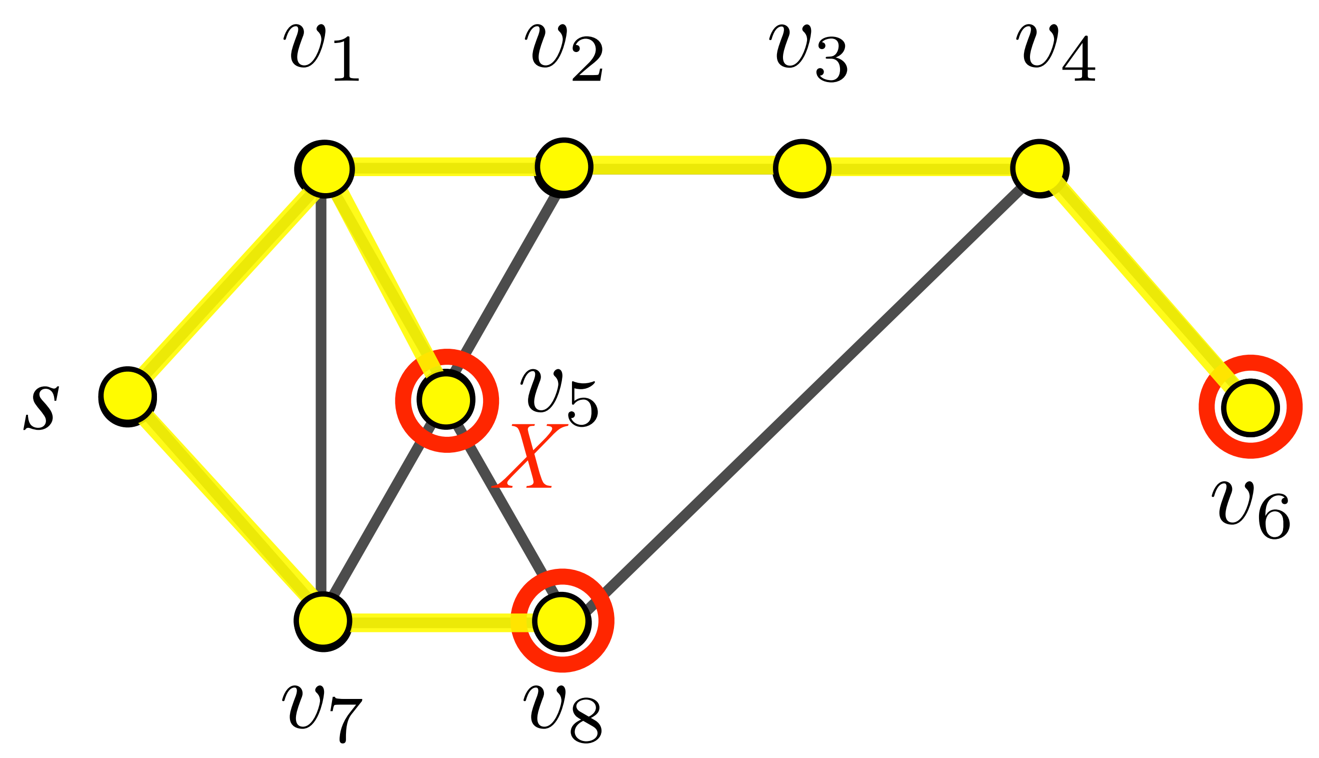
Algorithmus 3.7

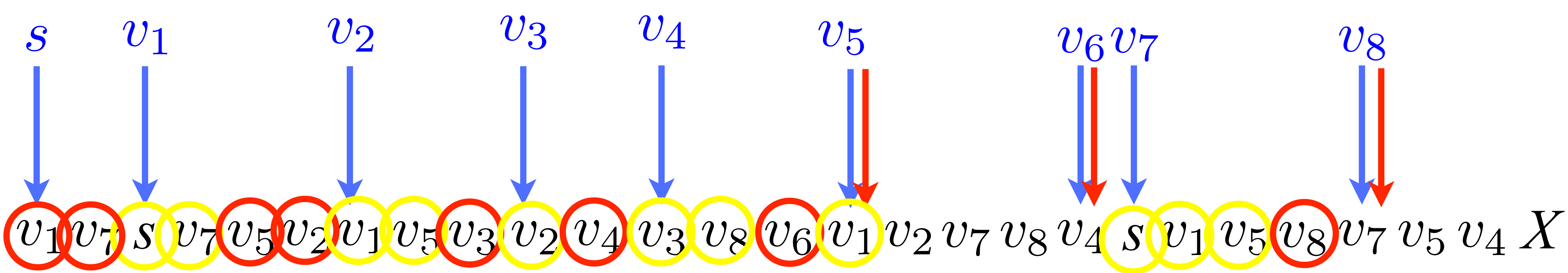
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

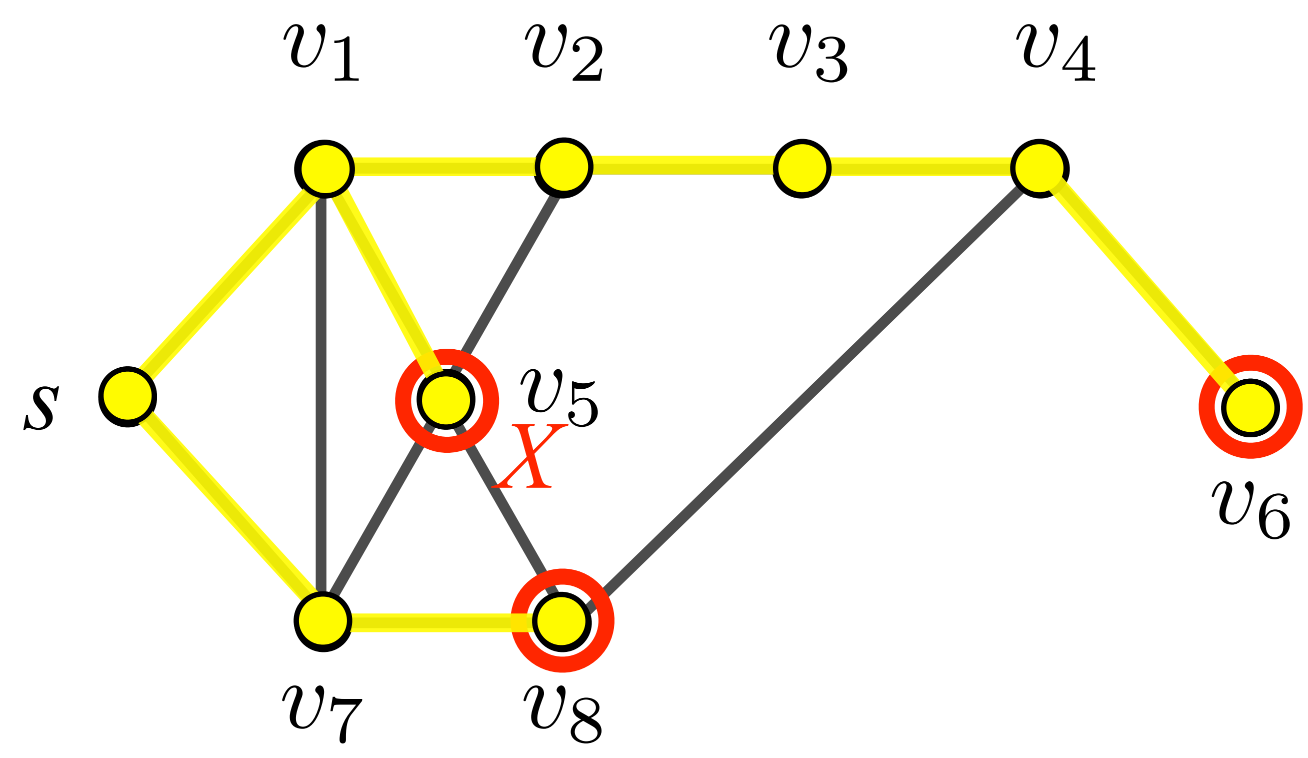


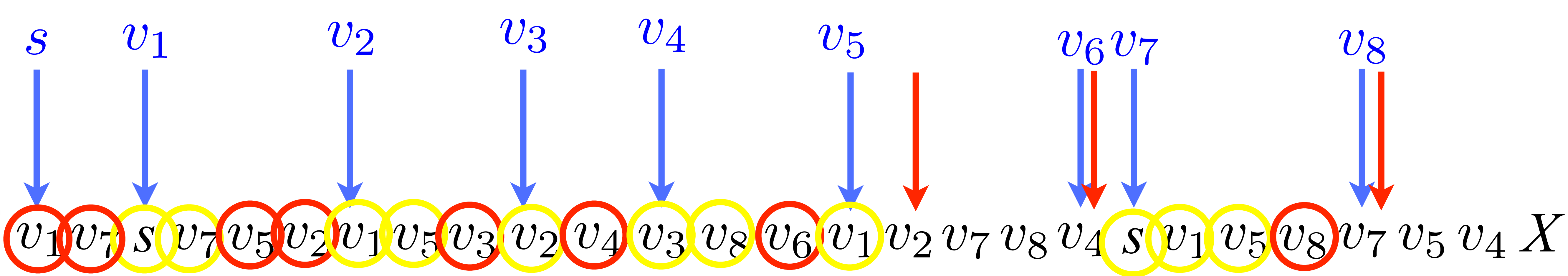


Algorithmus 3.7

INPUT:	Graph $G = (V,E)$, Knoten s
OUTPUT:	Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
	Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R:=\{s\}$, $Y:=\{s\}$, $T:=\emptyset$
2. WHILE ($R \neq \emptyset$) DO {
   2.1. Wähle $v \in R$
   2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
      2.2.1. $R:=R \setminus \{v\}$
   2.3. ELSE {
      2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
      2.3.2. Setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$
      }
   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

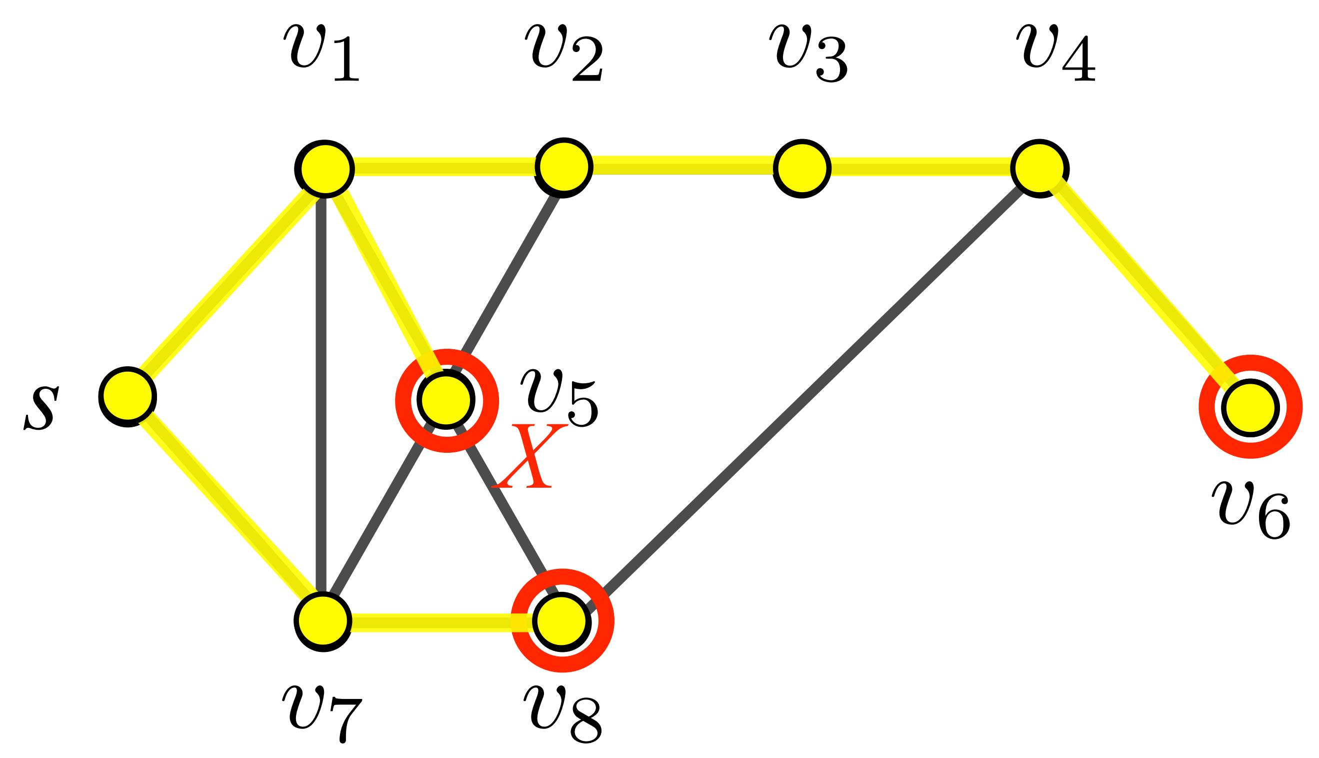
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

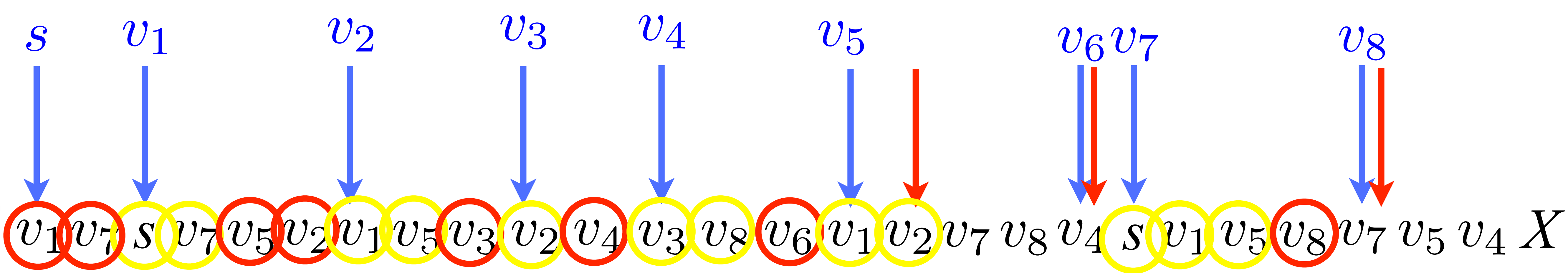
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





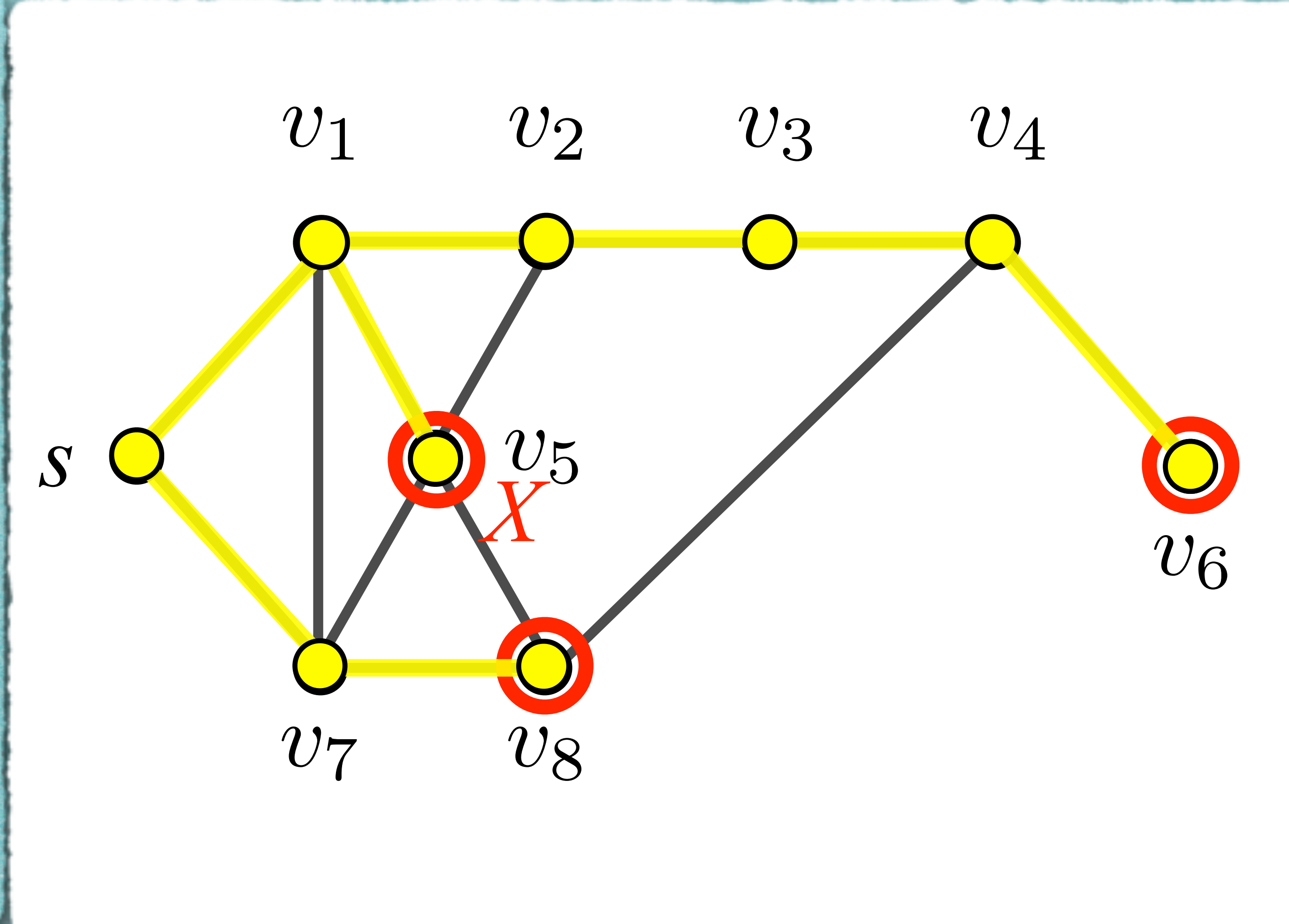
Algorithmus 3.7

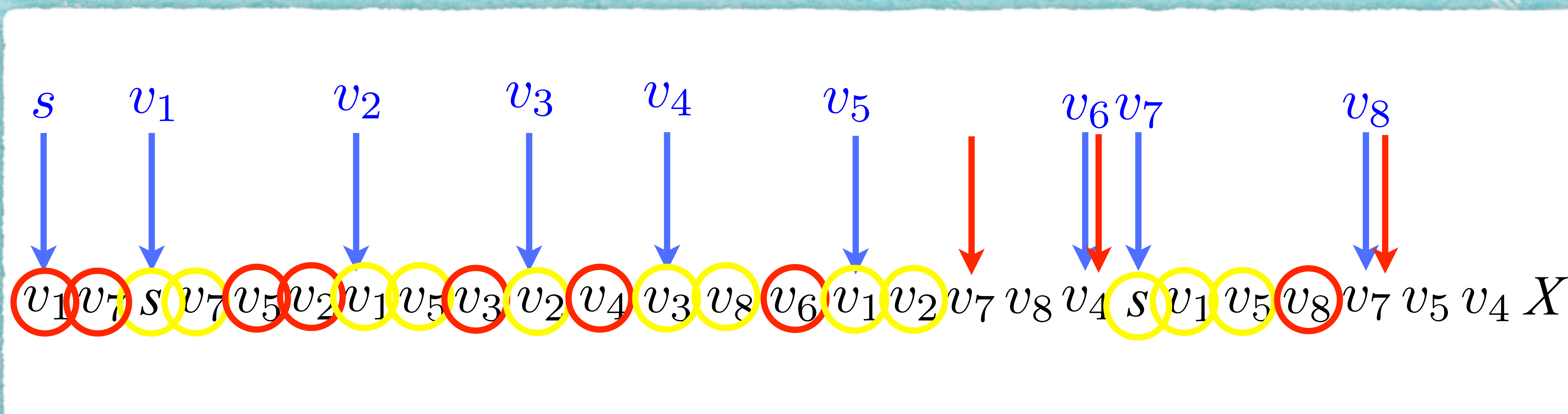
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

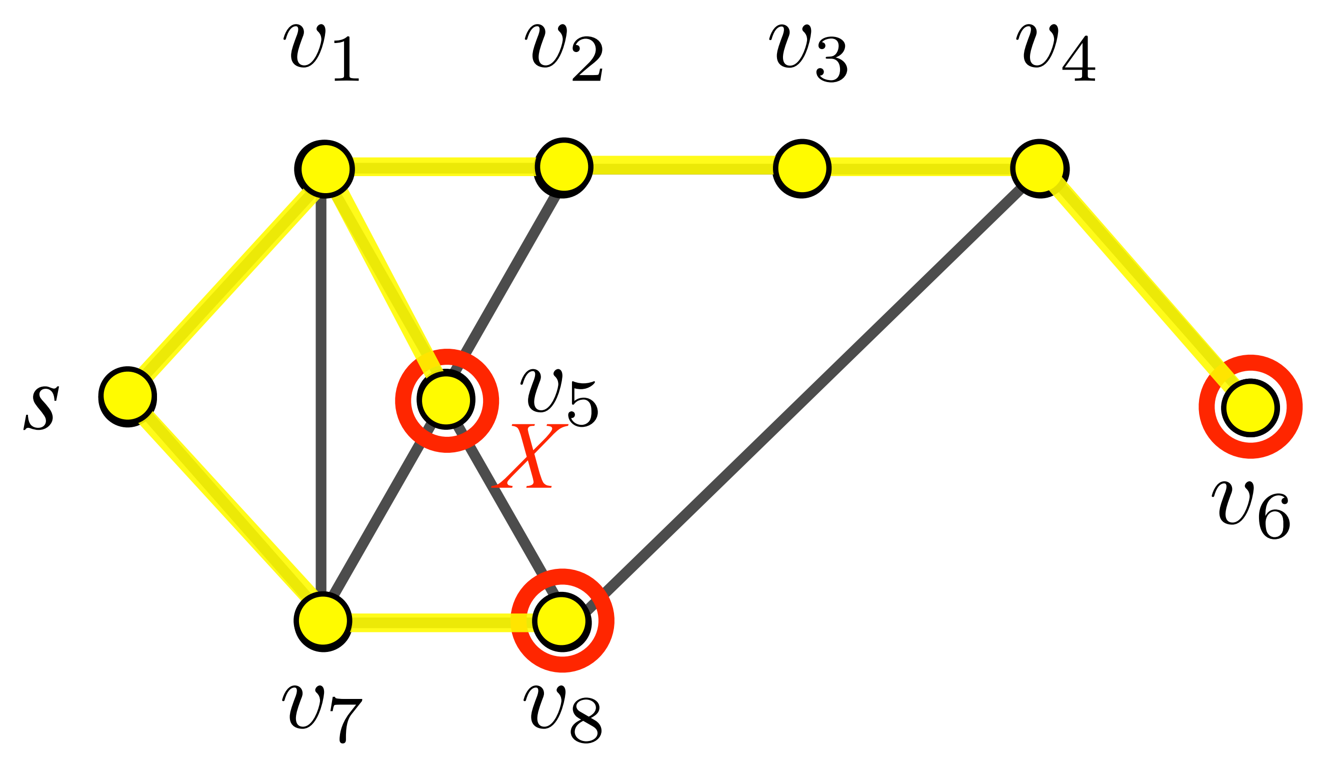
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

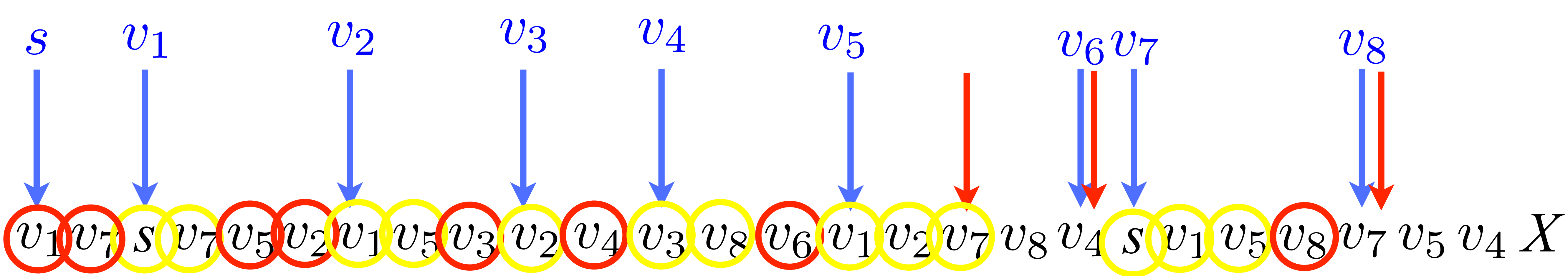
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

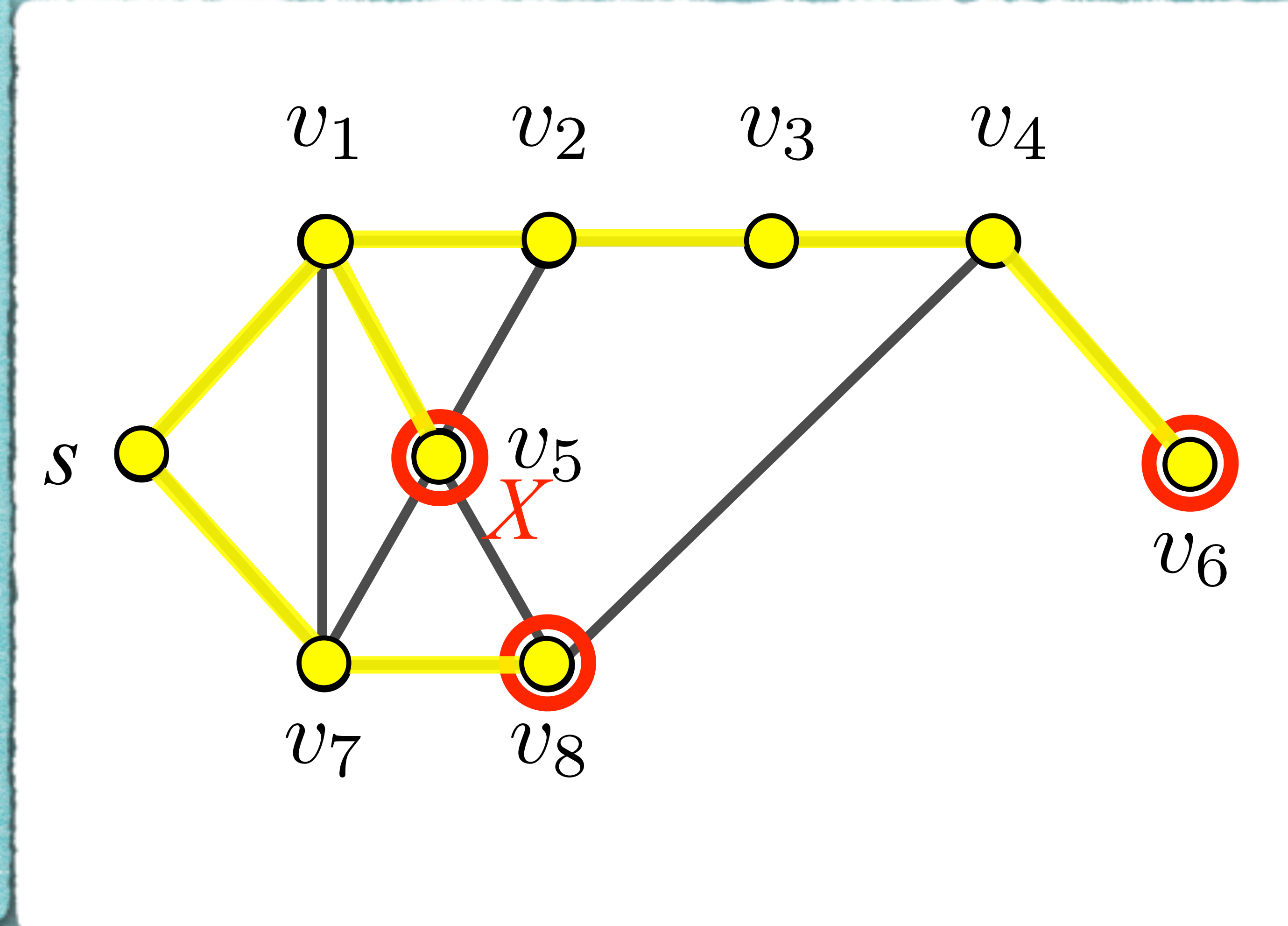


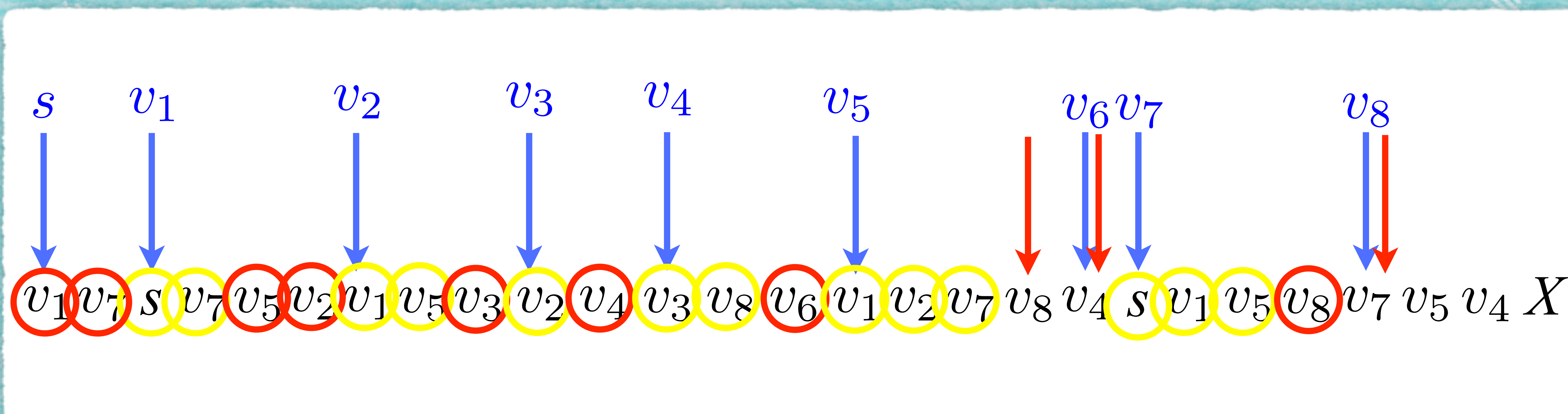


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP





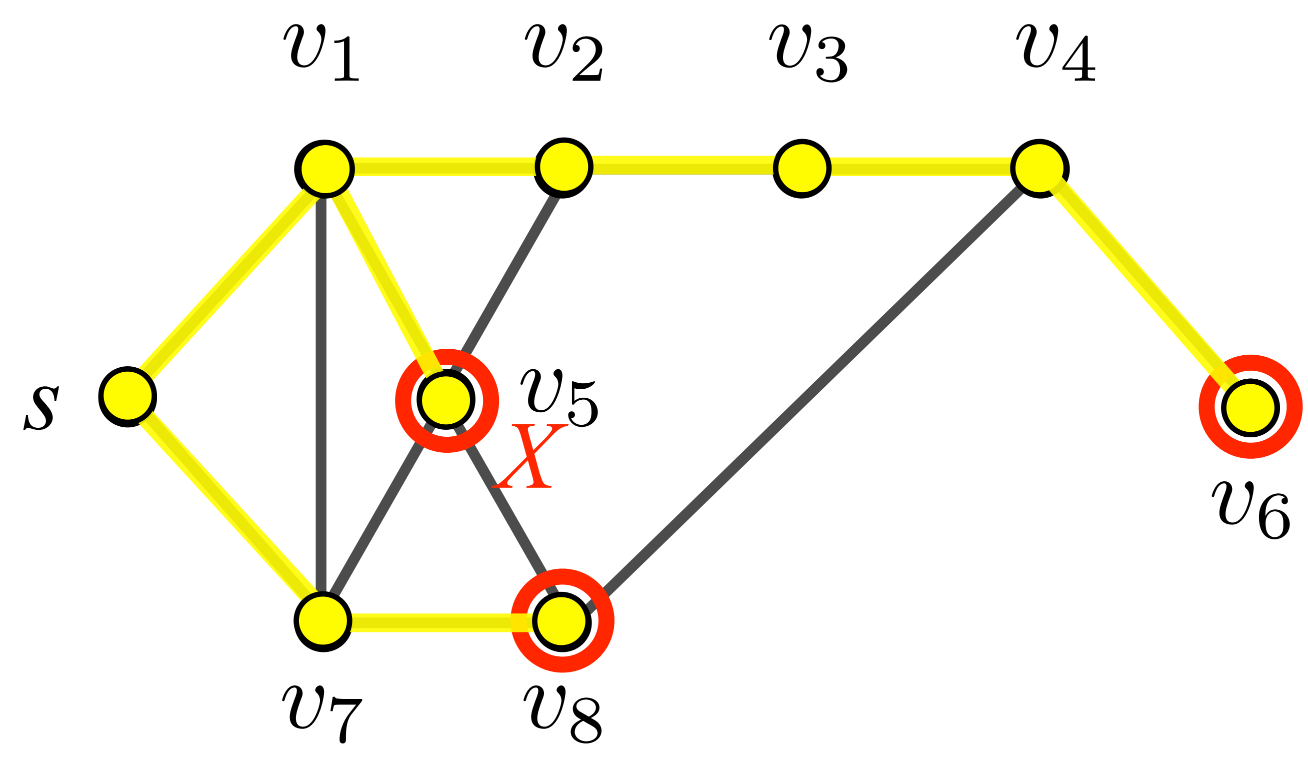
Algorithmus 3.7

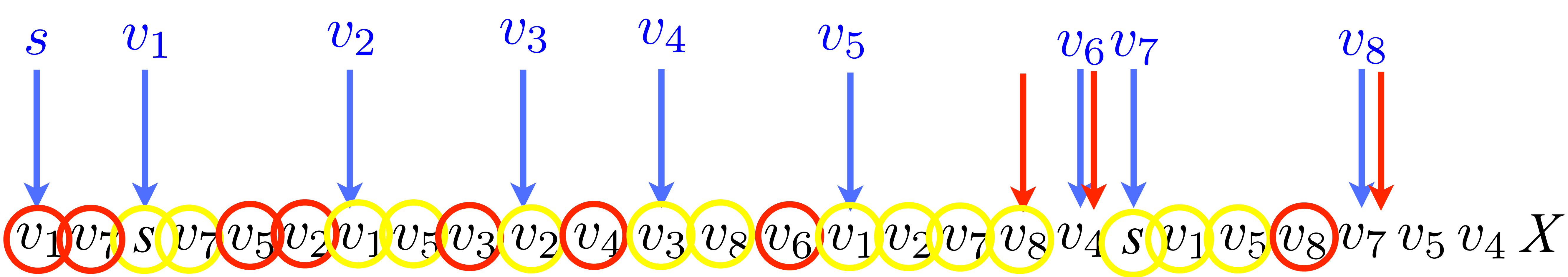
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP

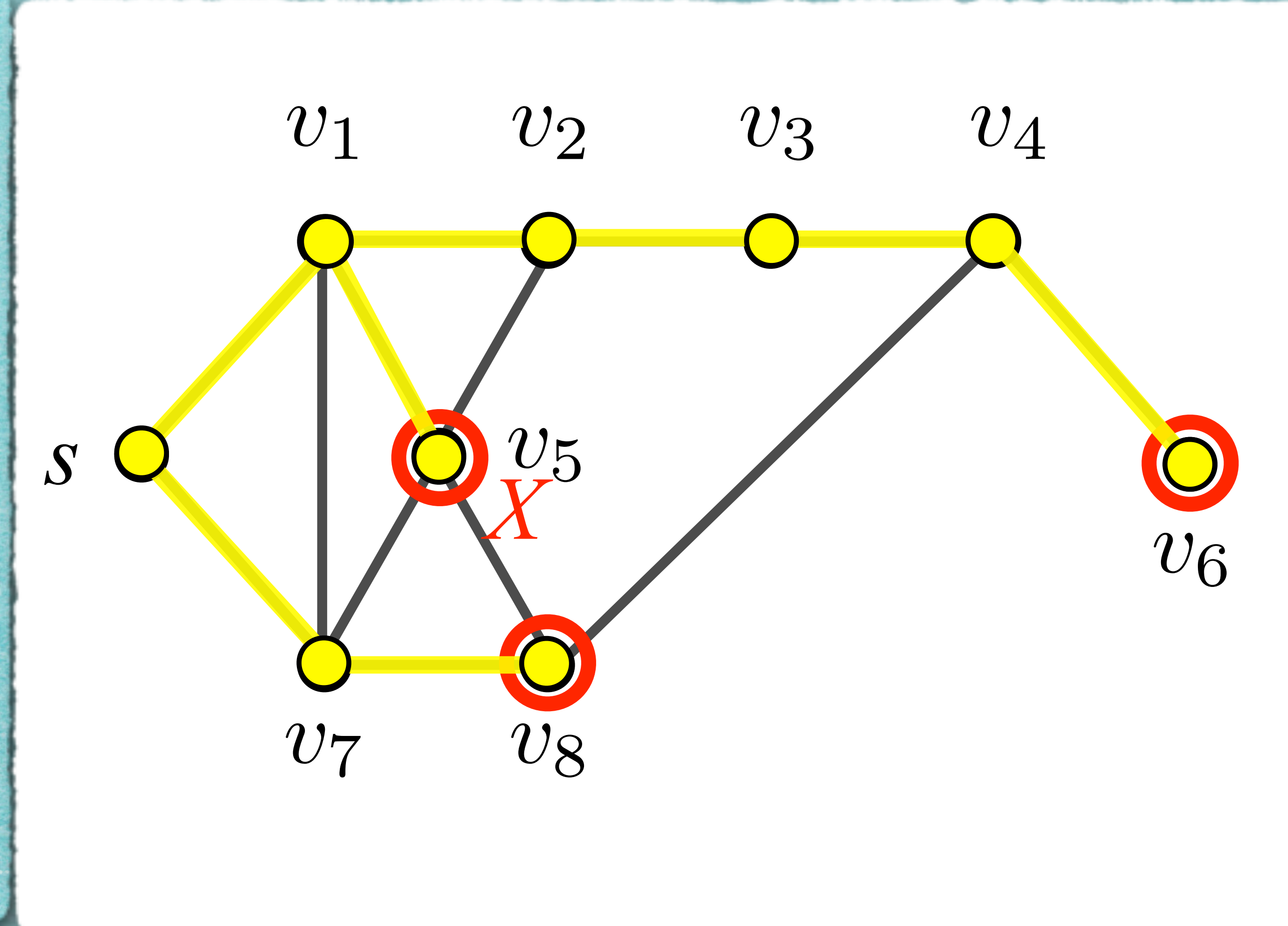


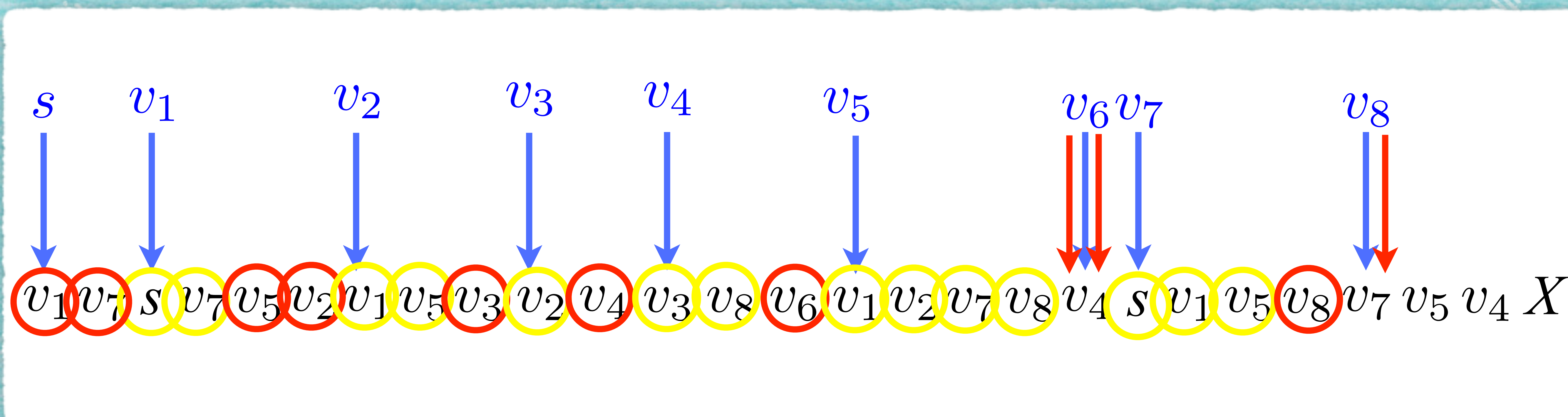


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
   }
3. STOP





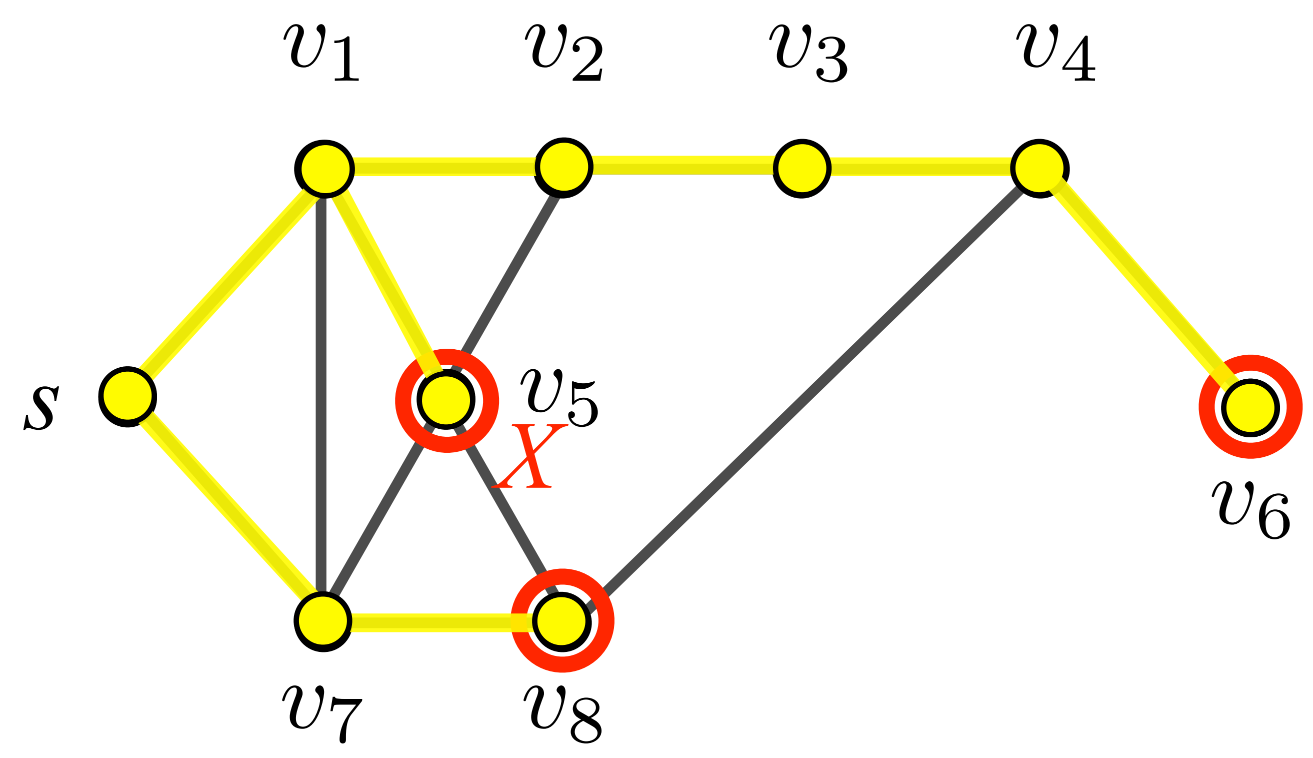
Algorithmus 3.7

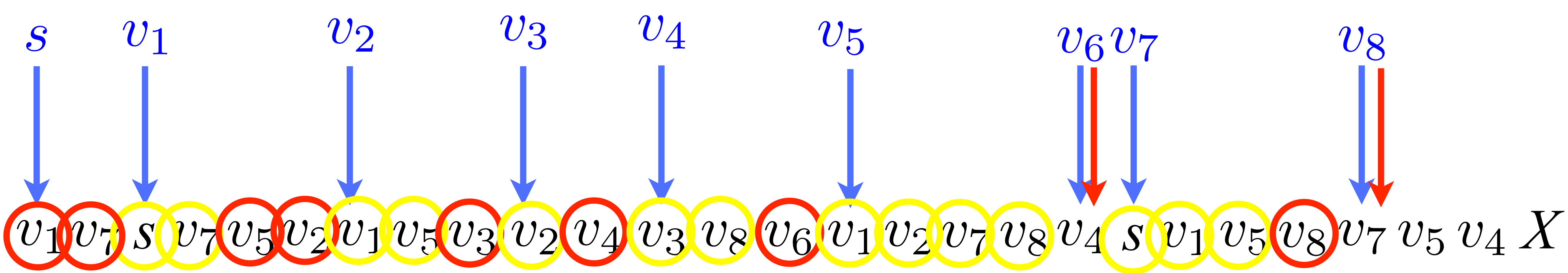
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈ E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





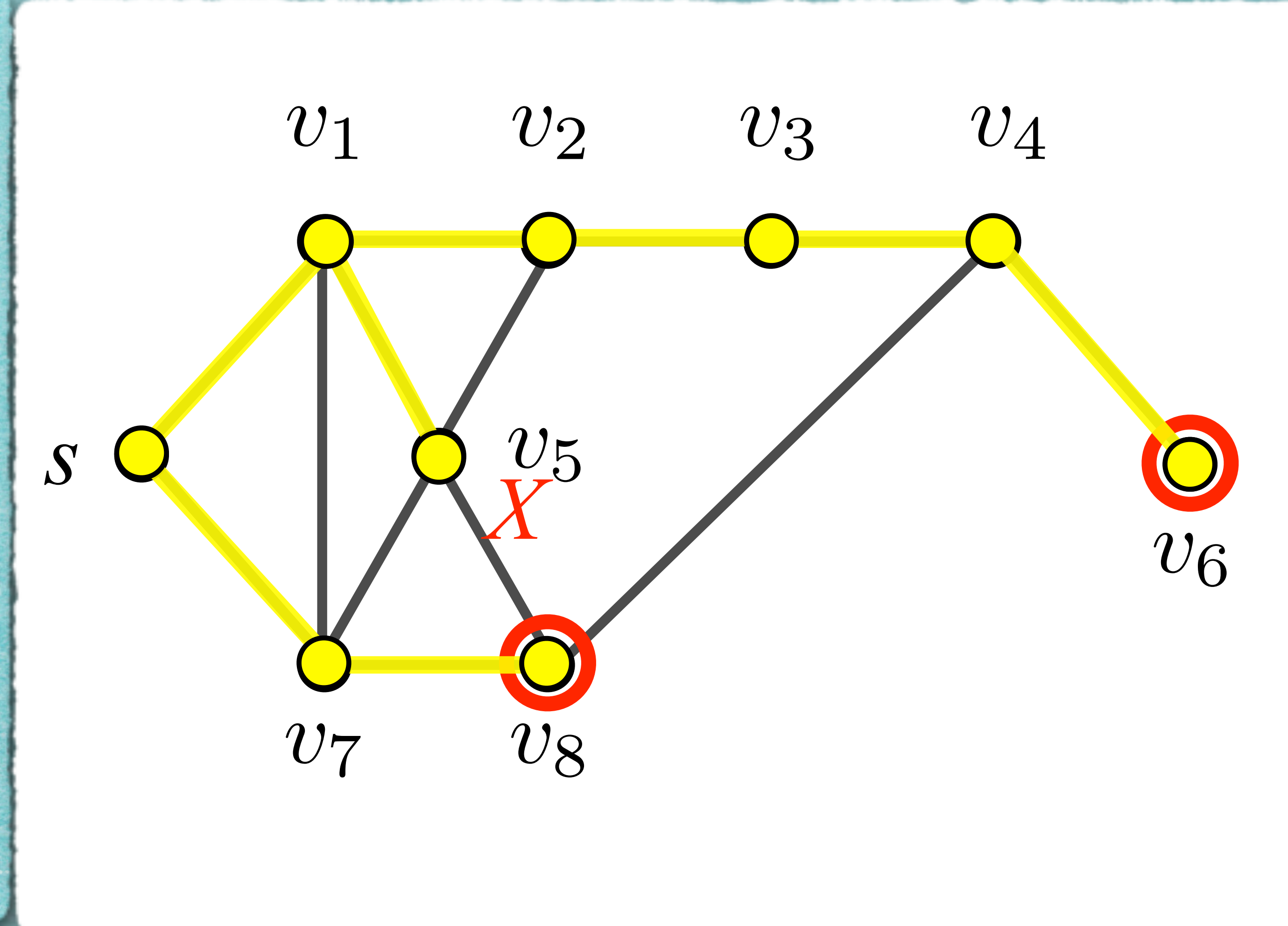
Algorithmus 3.7

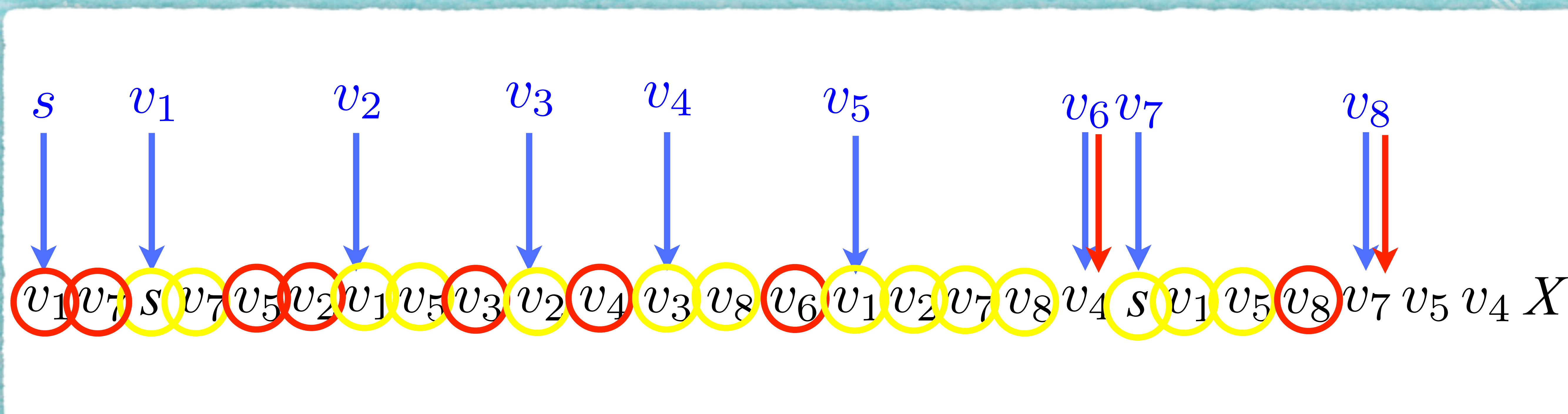
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }
   - }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

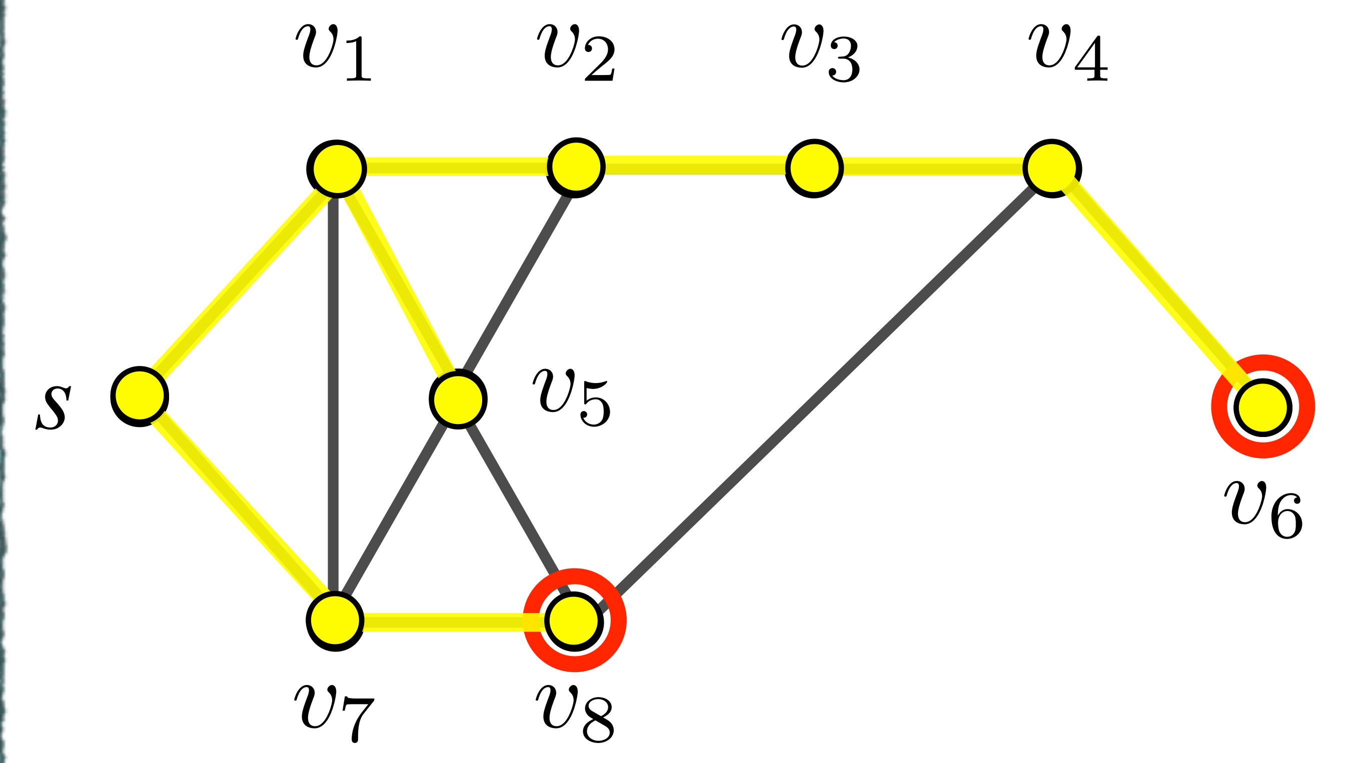
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

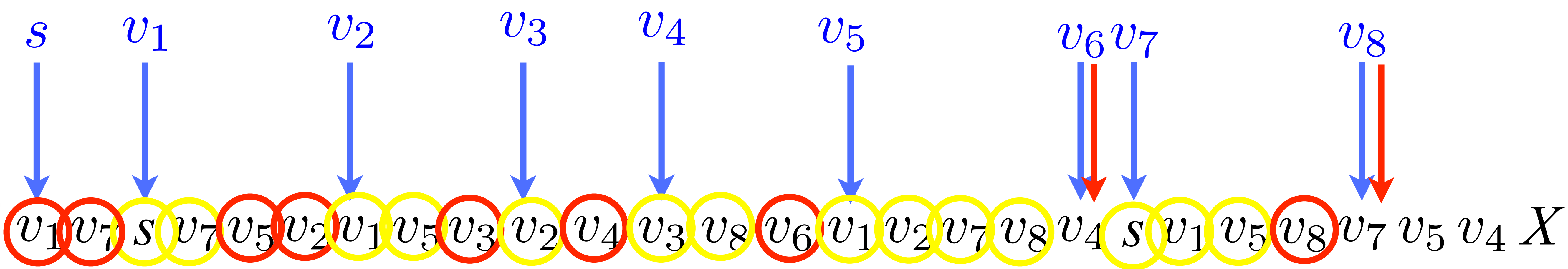
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

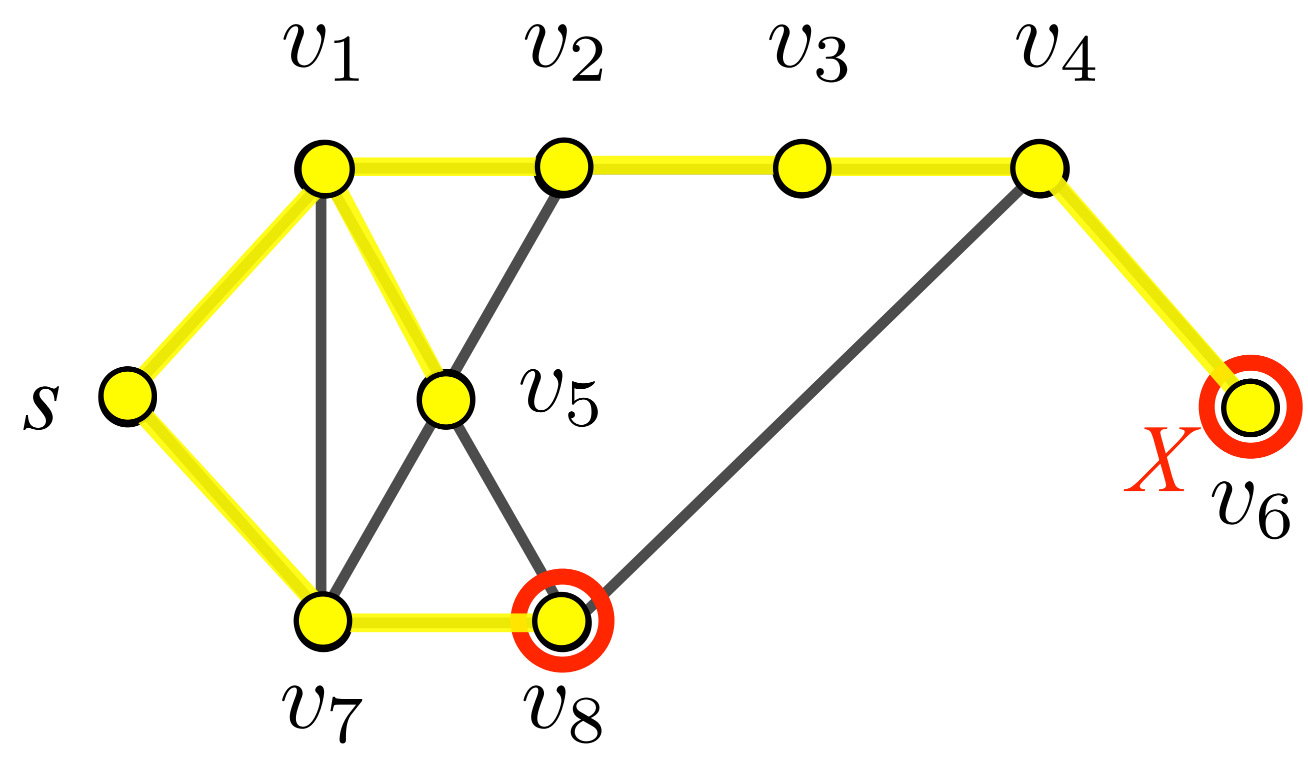
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

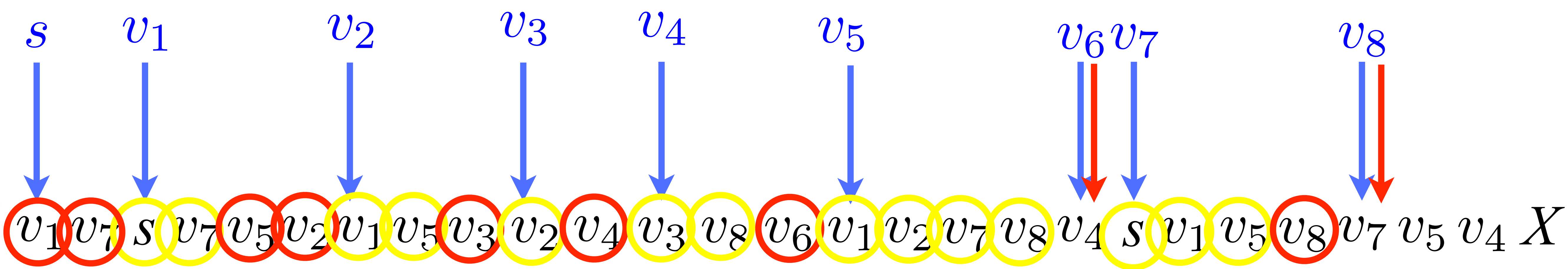
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

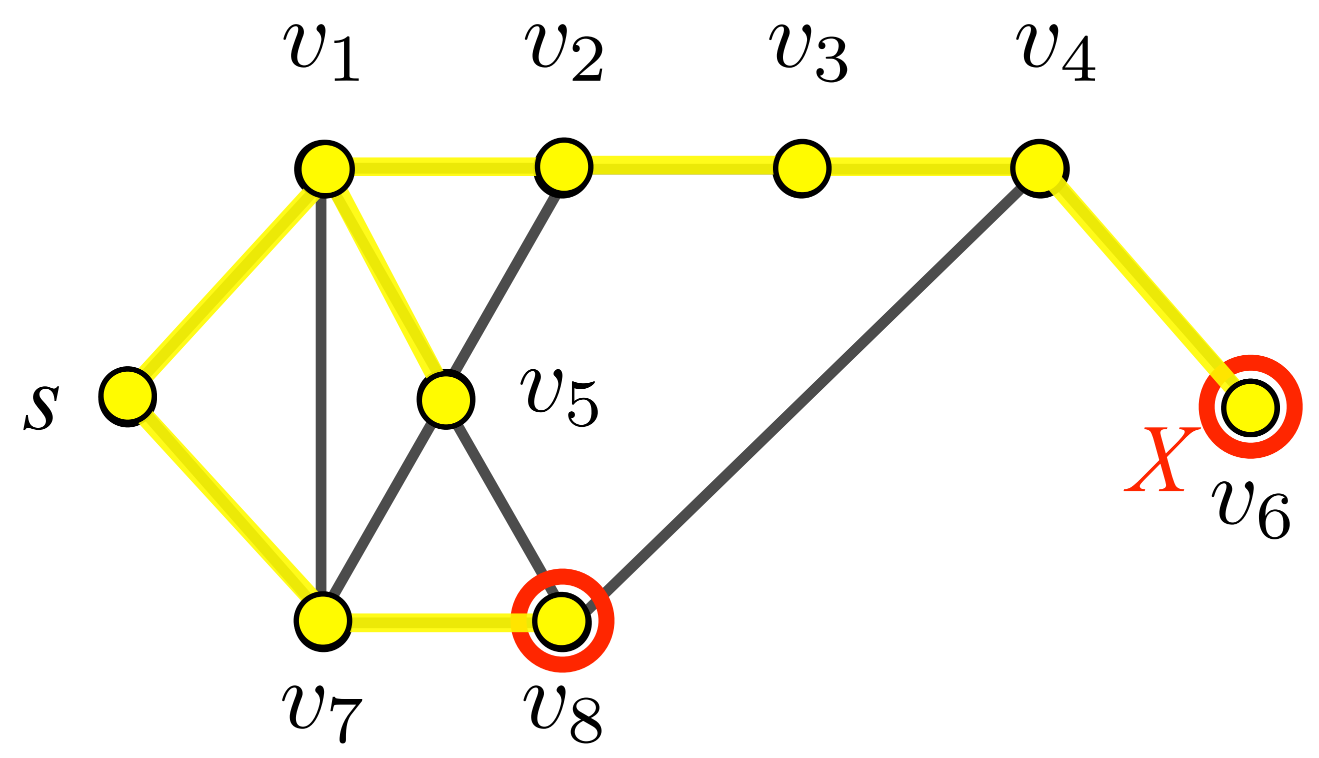
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

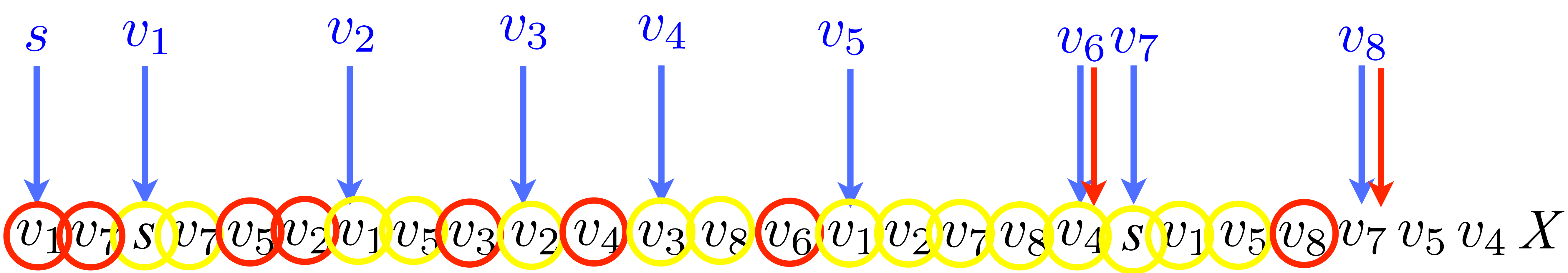
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}

     }

   }
3. STOP





Algorithmus 3.7

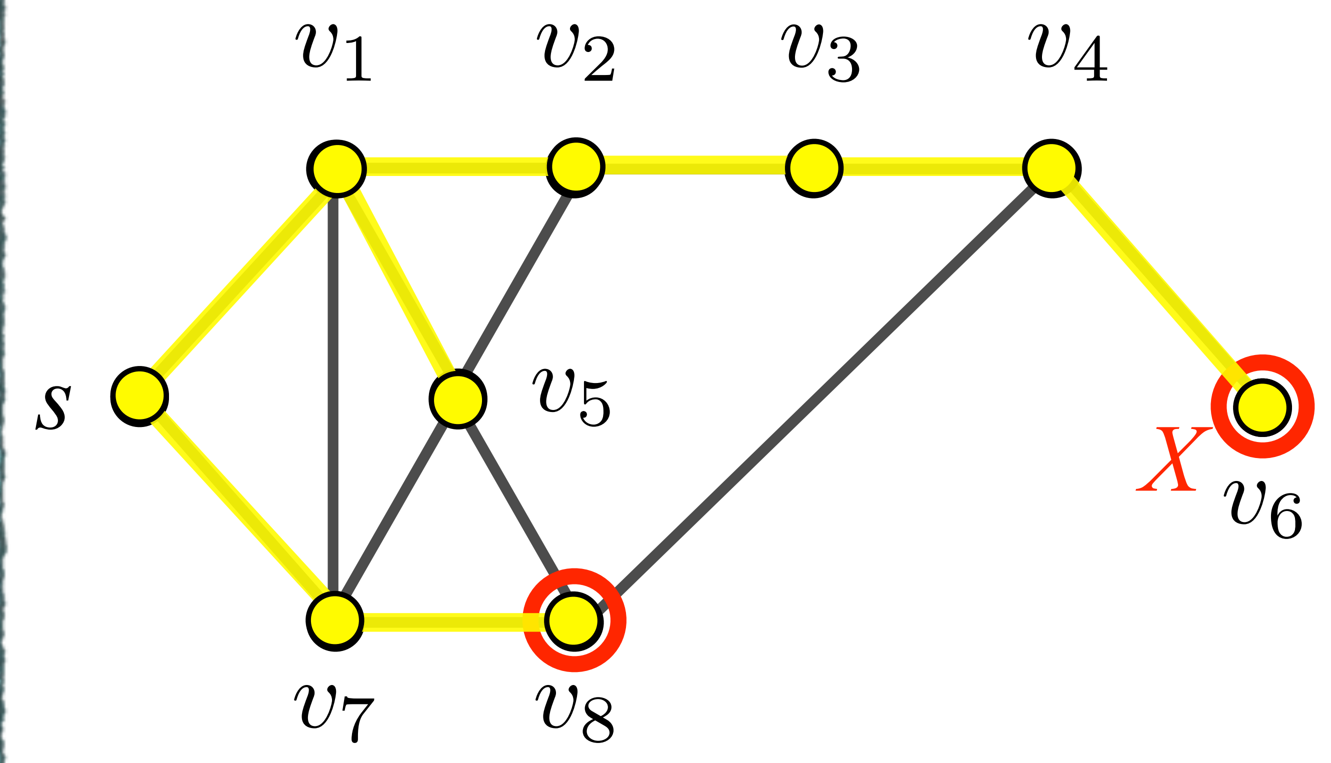
INPUT: Graph G = (V,E), Knoten s

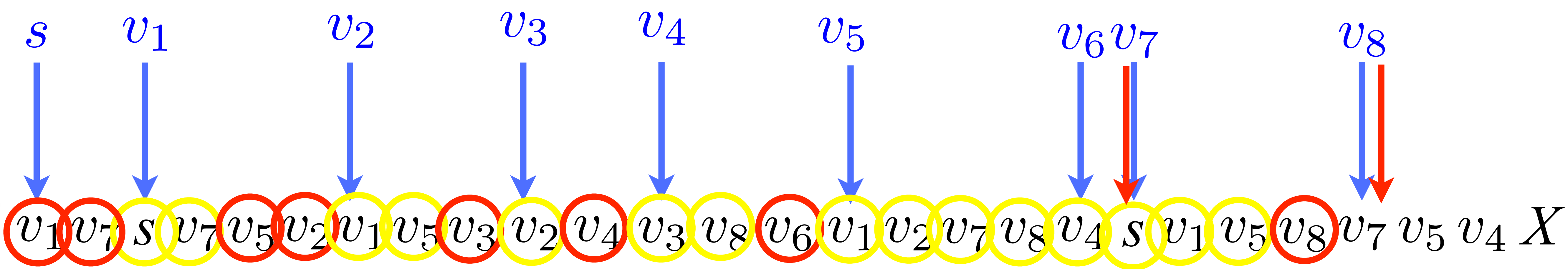
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        - }

    }
3. STOP



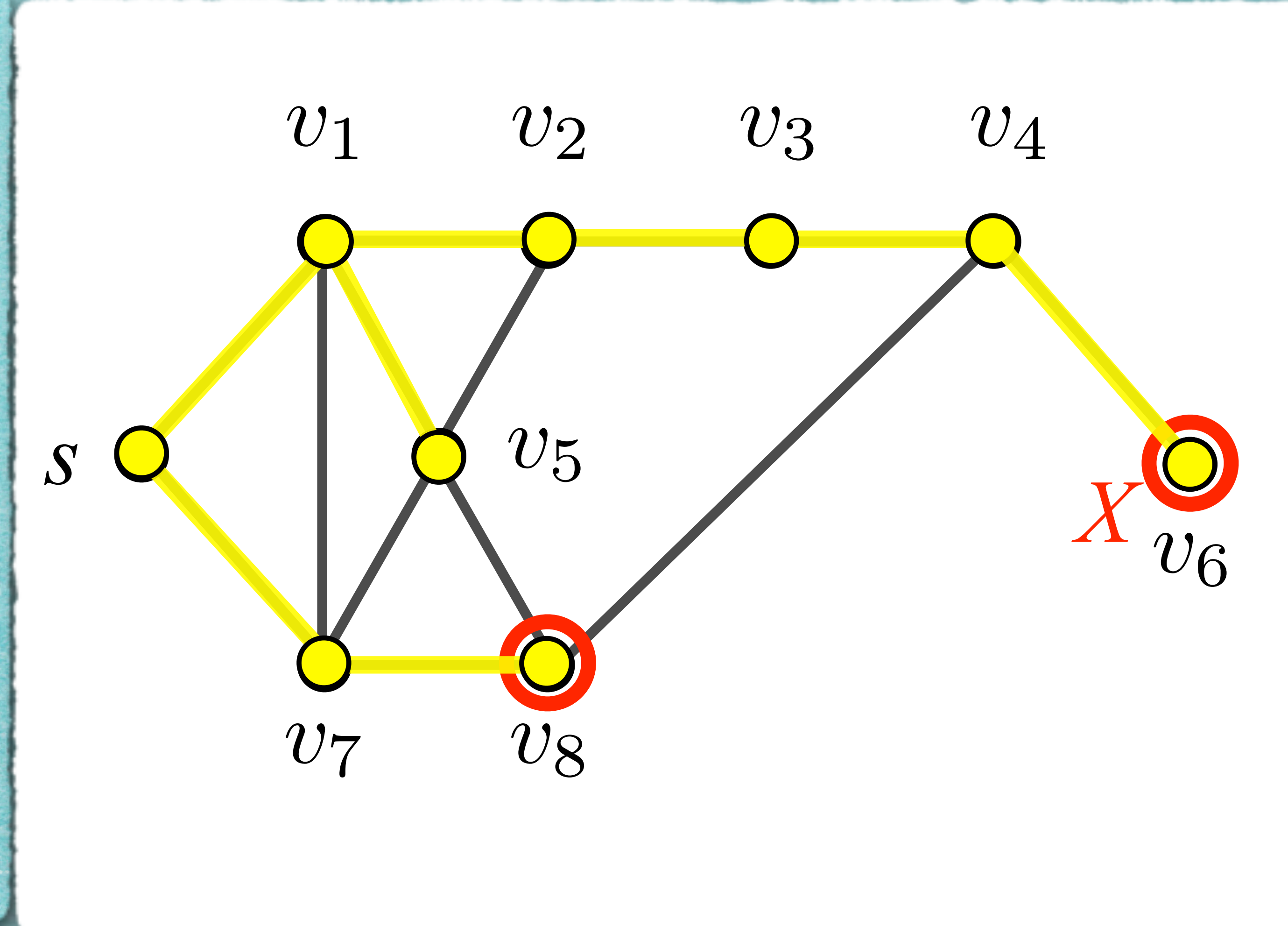


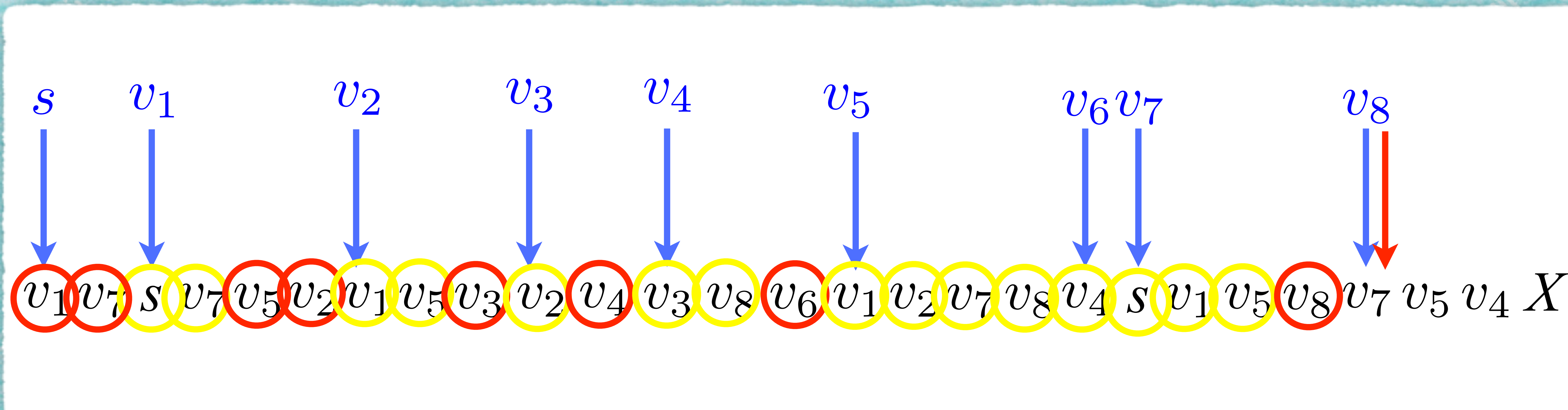
Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP

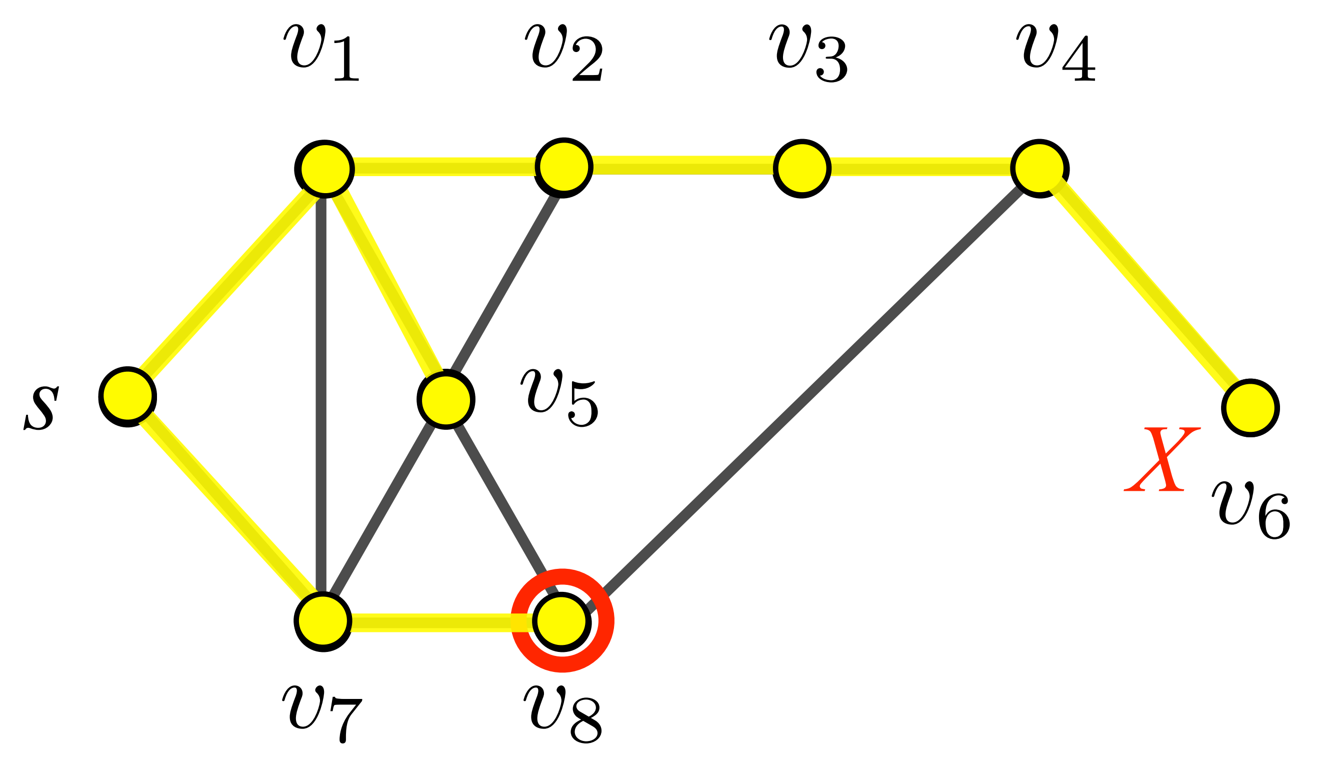


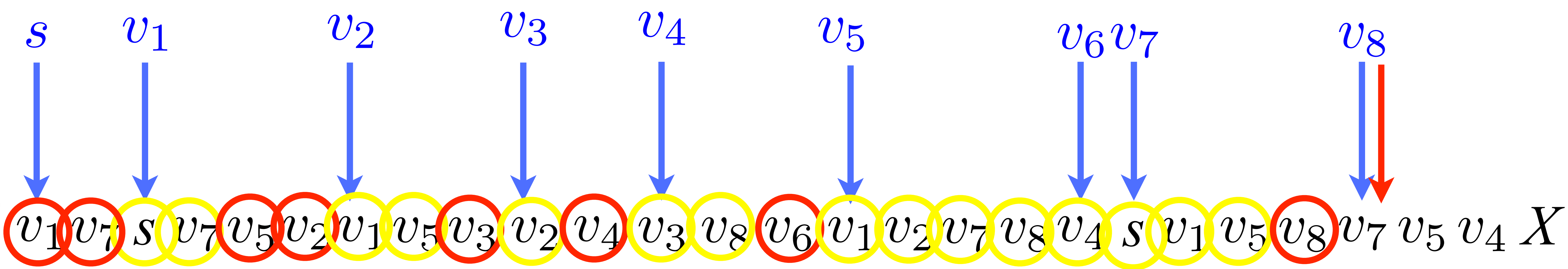


Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R ∪ {w}, Y := Y ∪ {w}, T := T ∪ {e}
        }
    }
3. STOP





Algorithmus 3.7

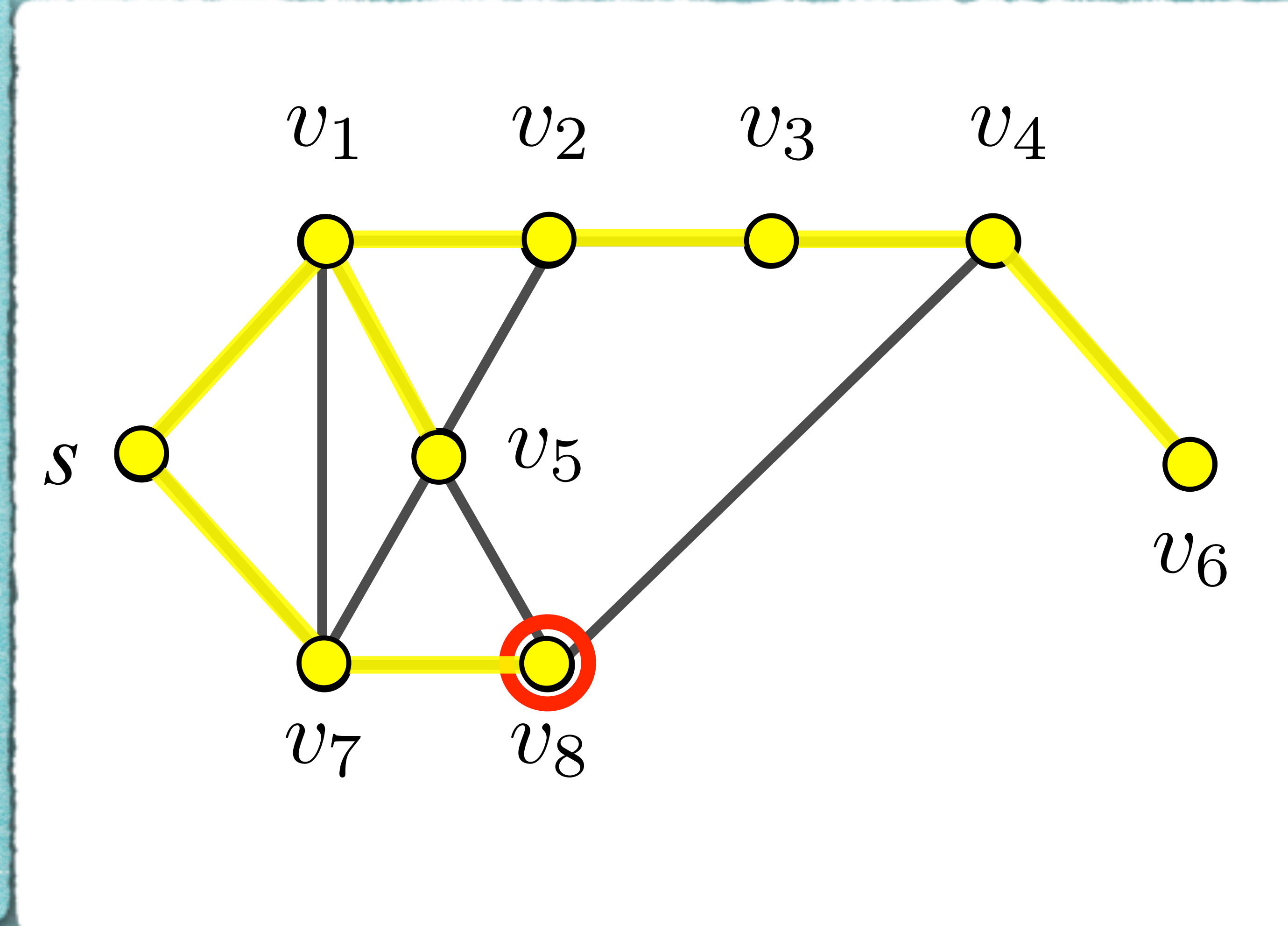
INPUT: Graph G = (V,E), Knoten s

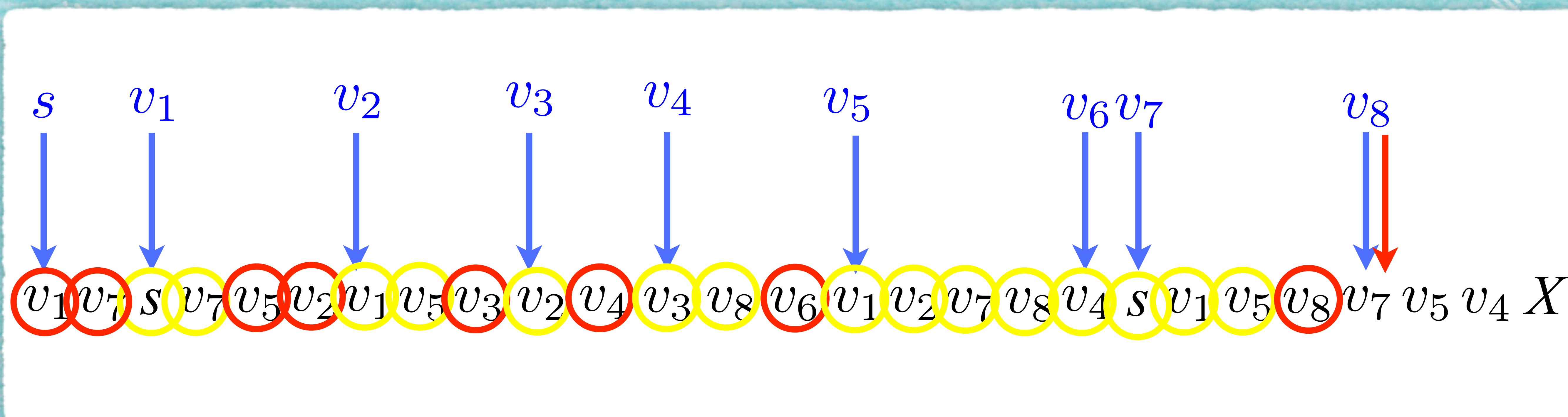
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

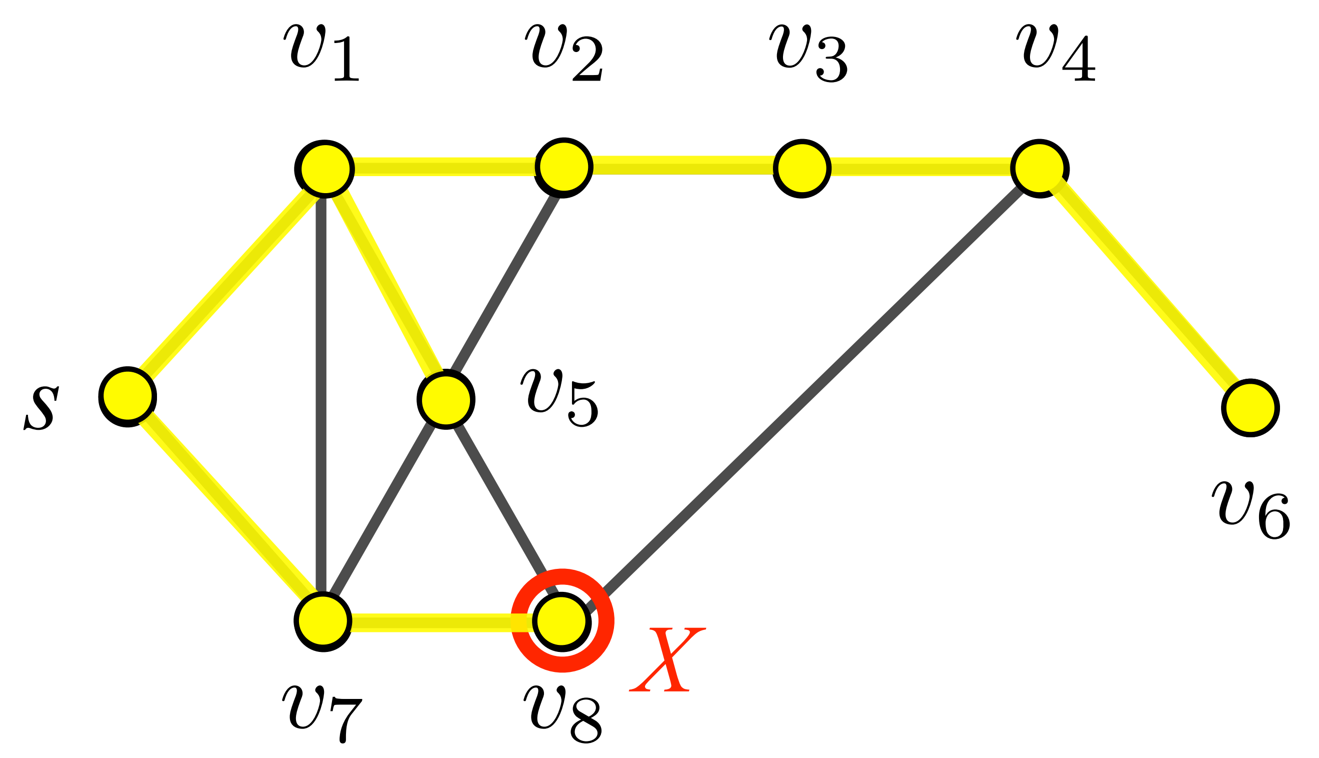
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

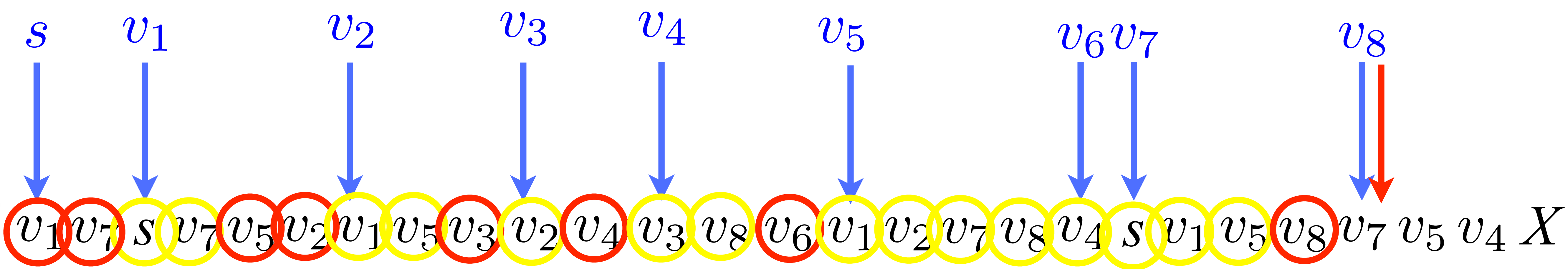
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w ∈ V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

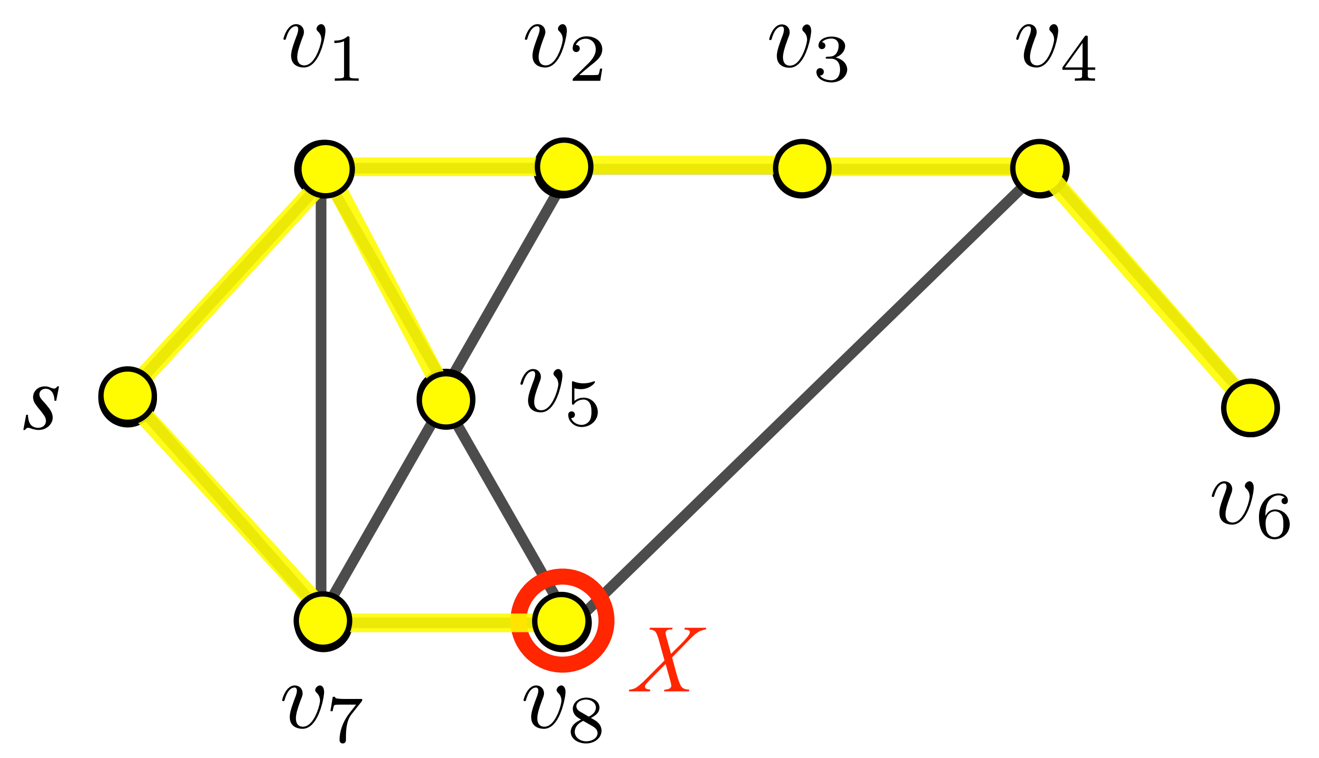
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

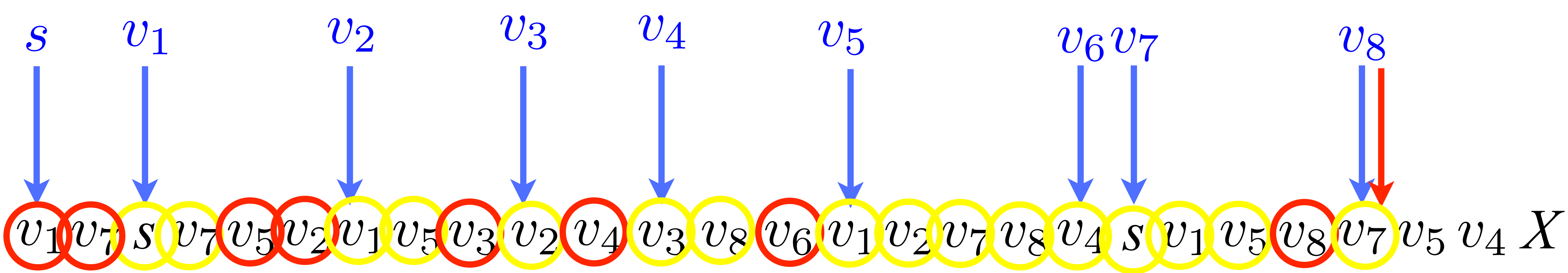
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {

   2.1. Wähle v∈ R

   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN

   2.2.1. R:=R\{v}

   2.3. ELSE {

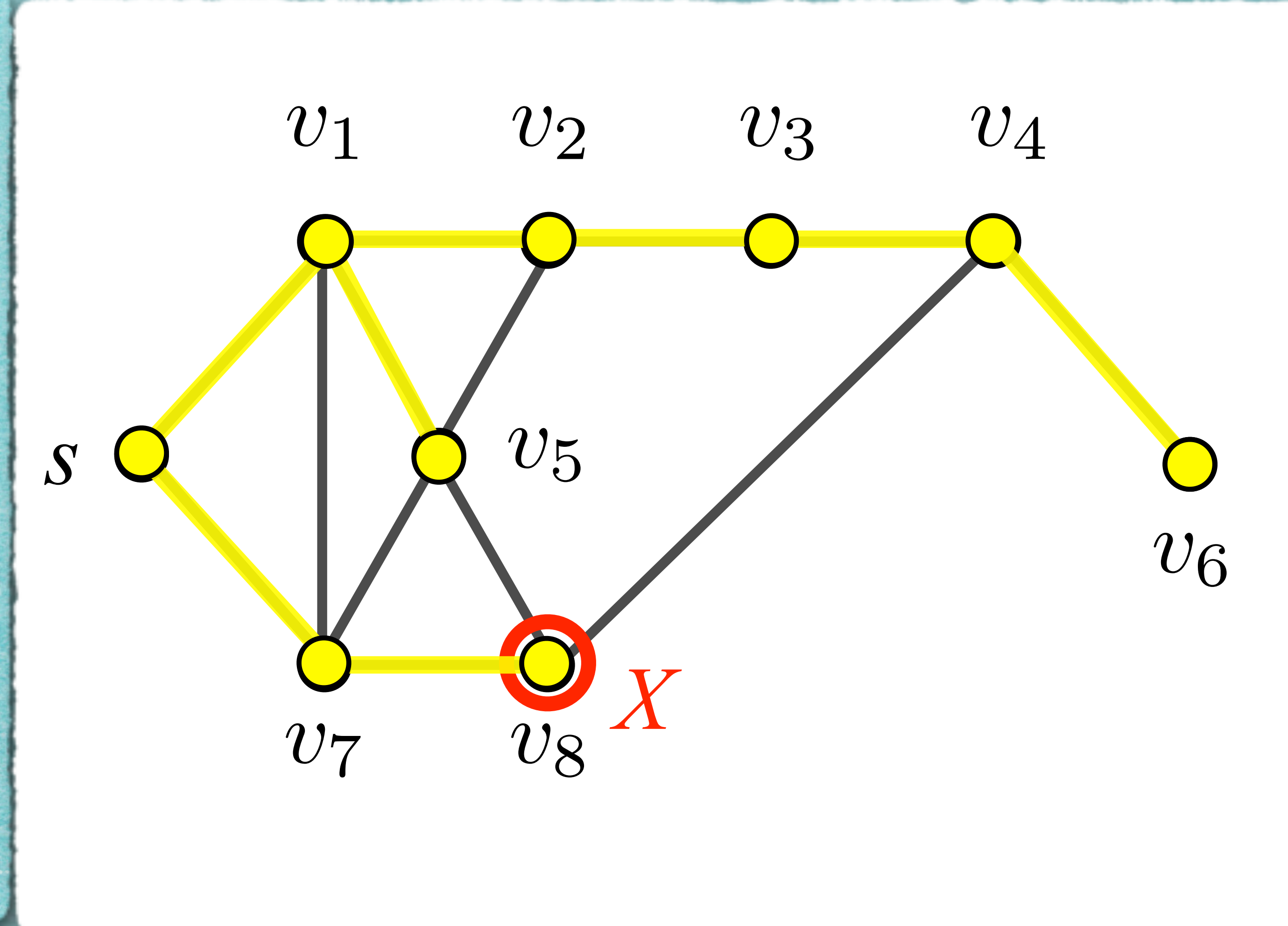
   2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

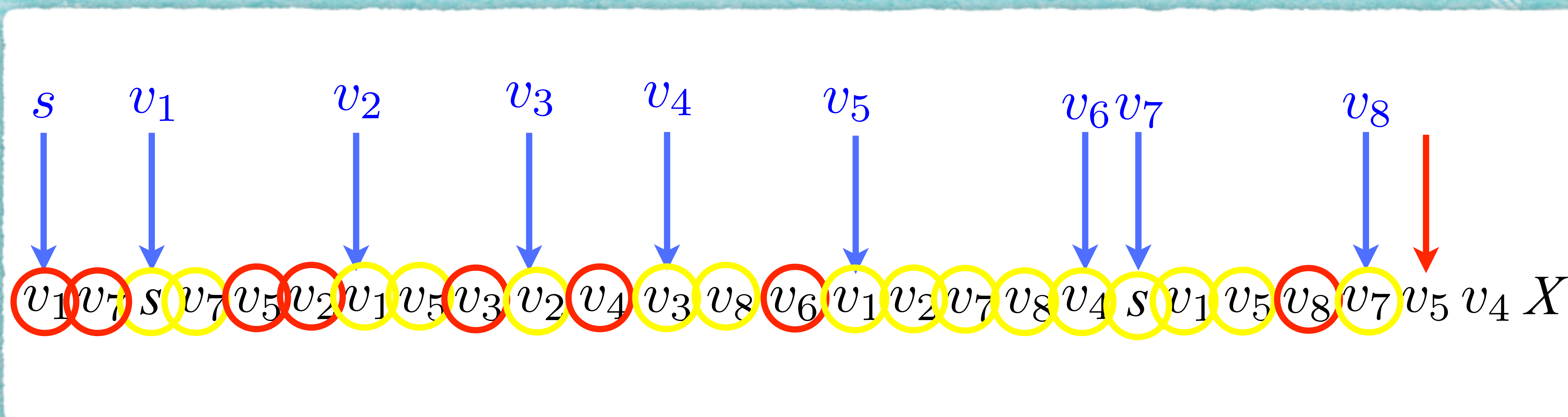
   2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

   }

   }

3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

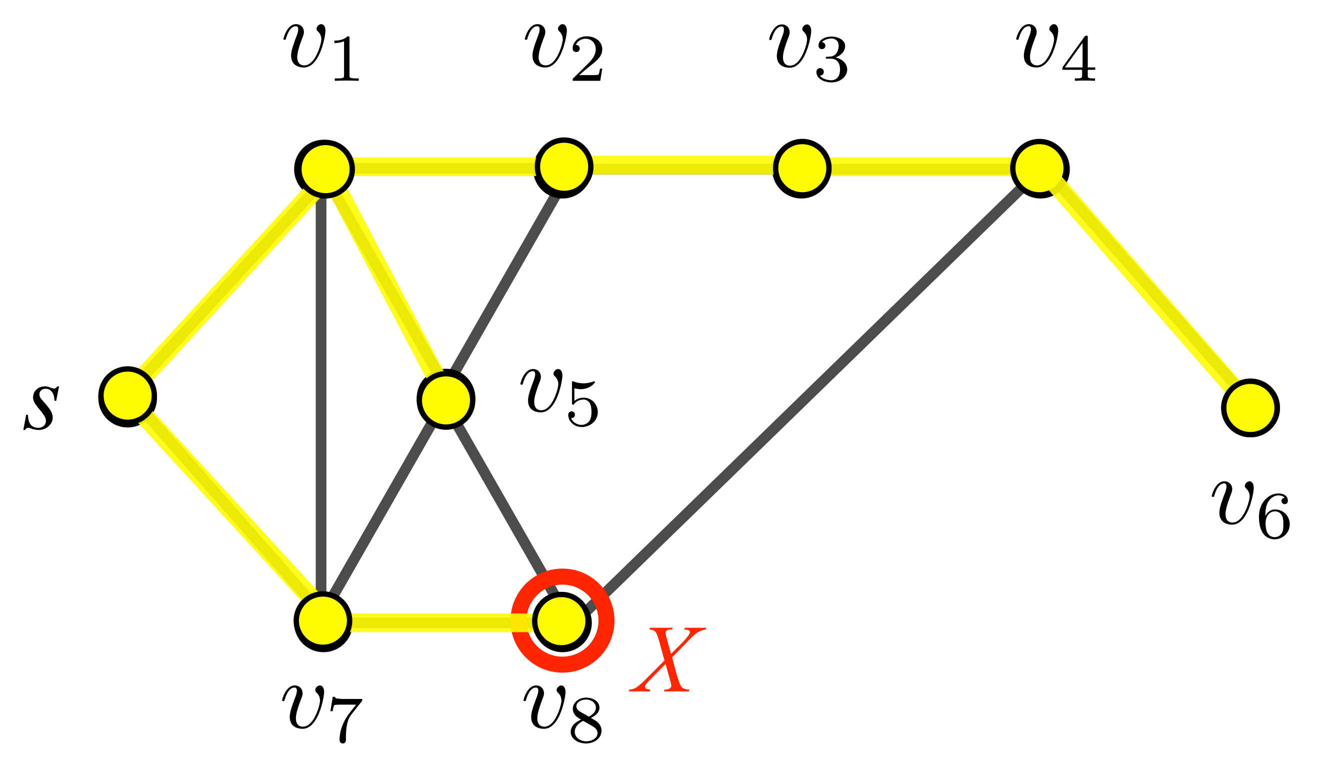
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

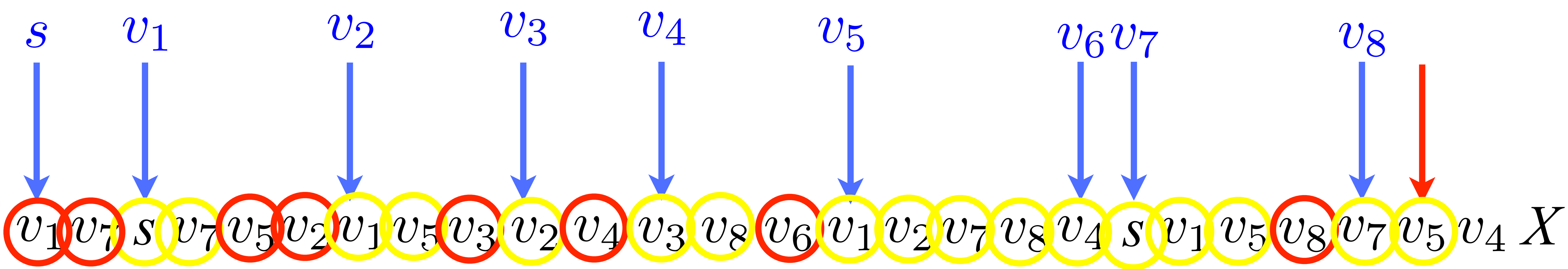
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





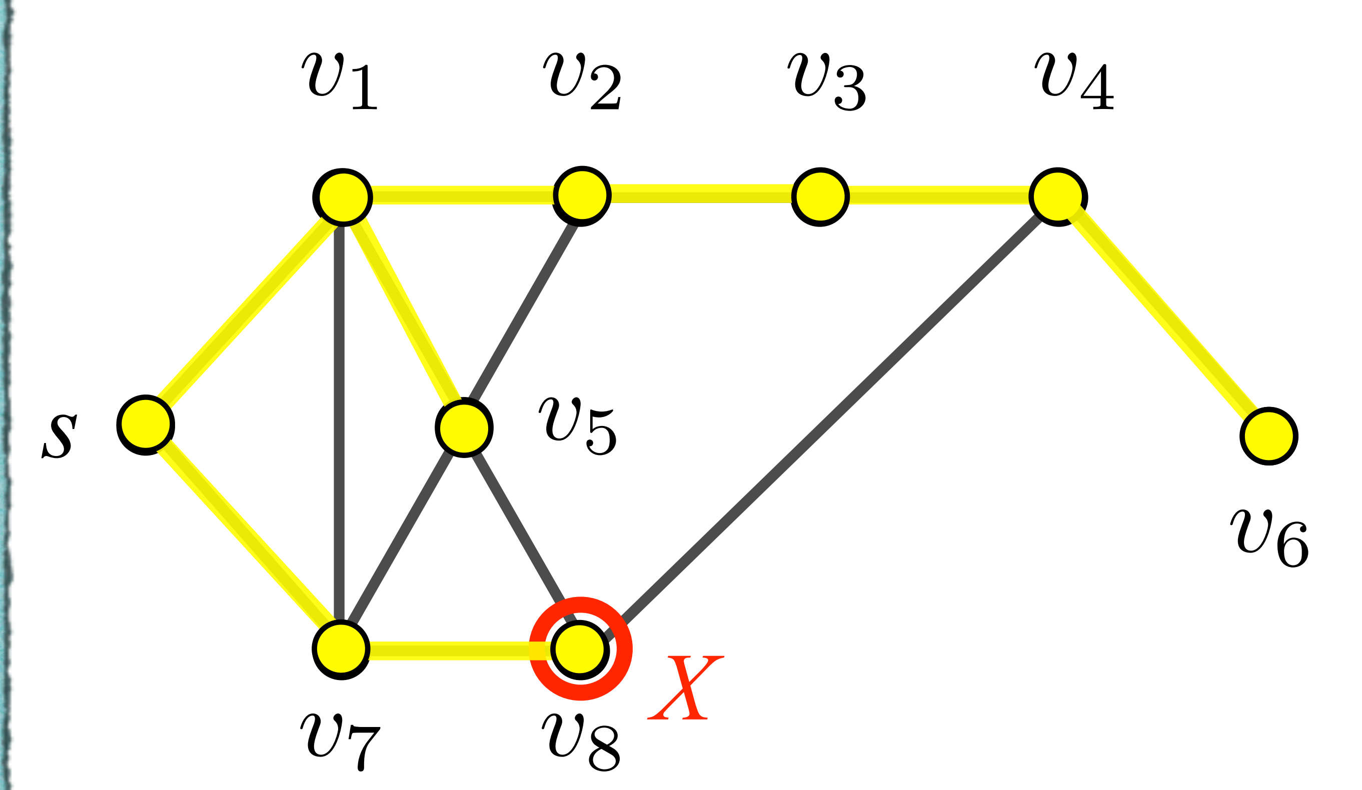
Algorithmus 3.7

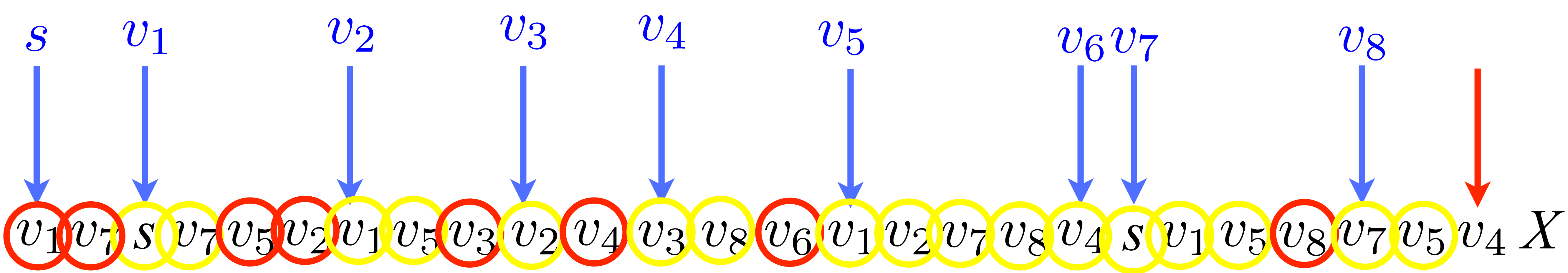
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

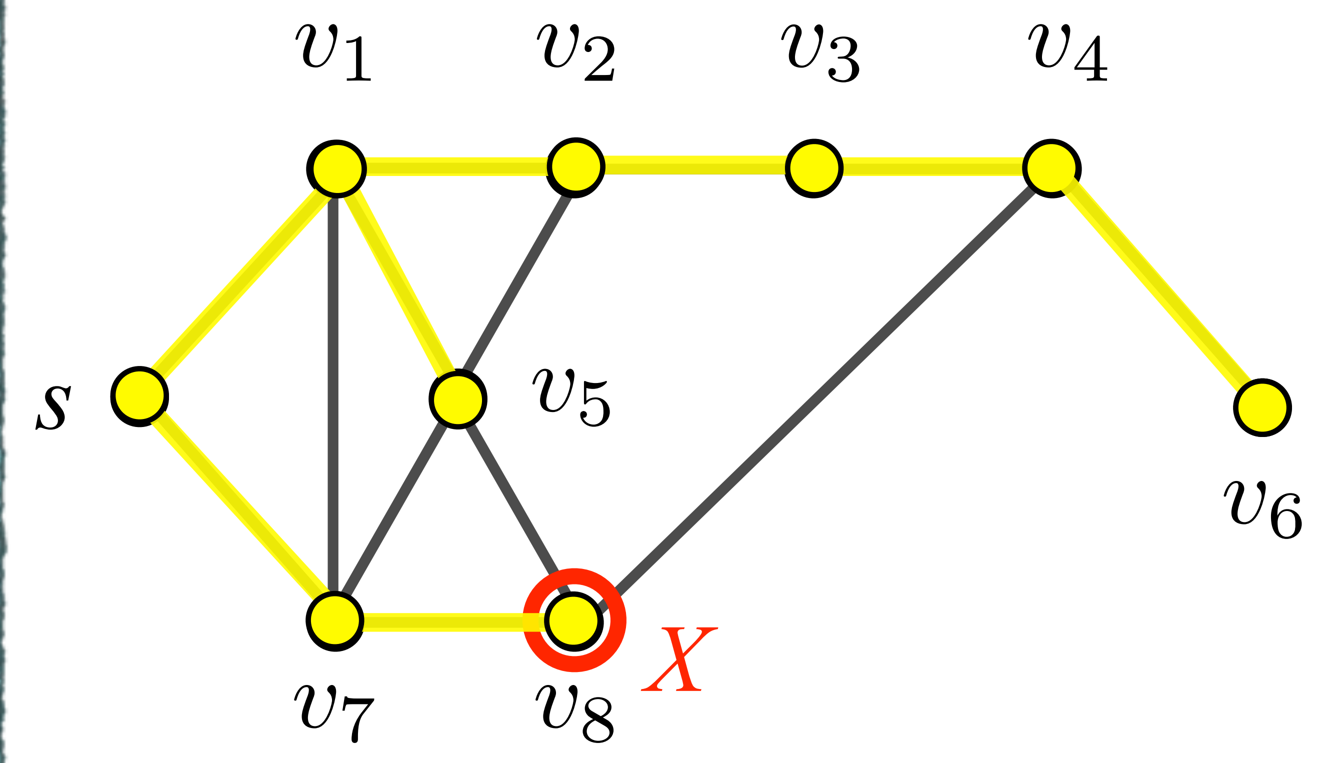
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

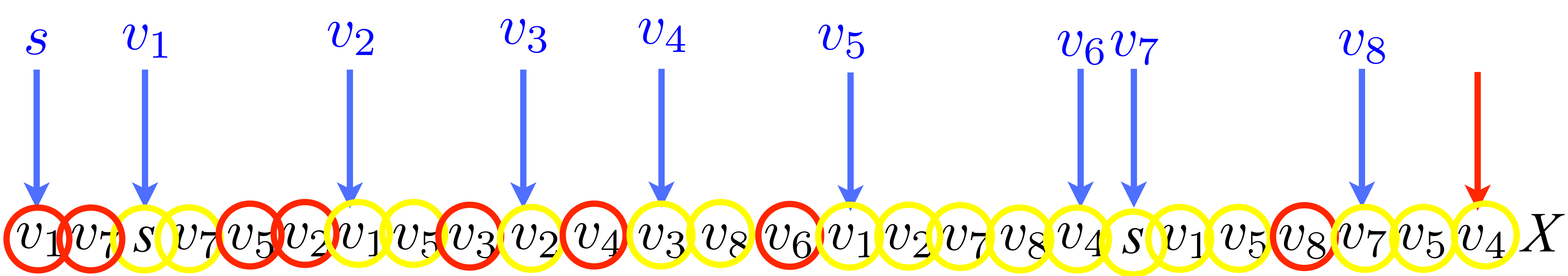
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





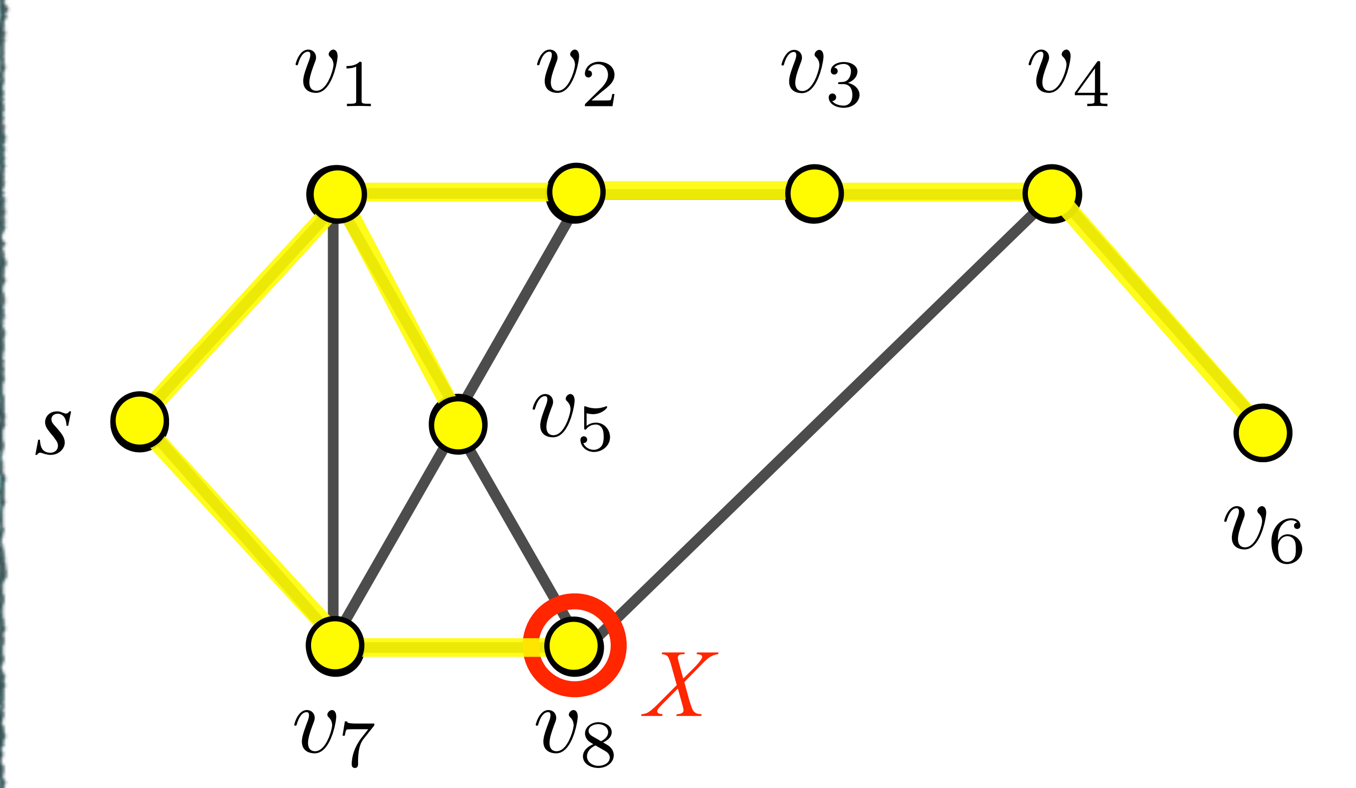
Algorithmus 3.7

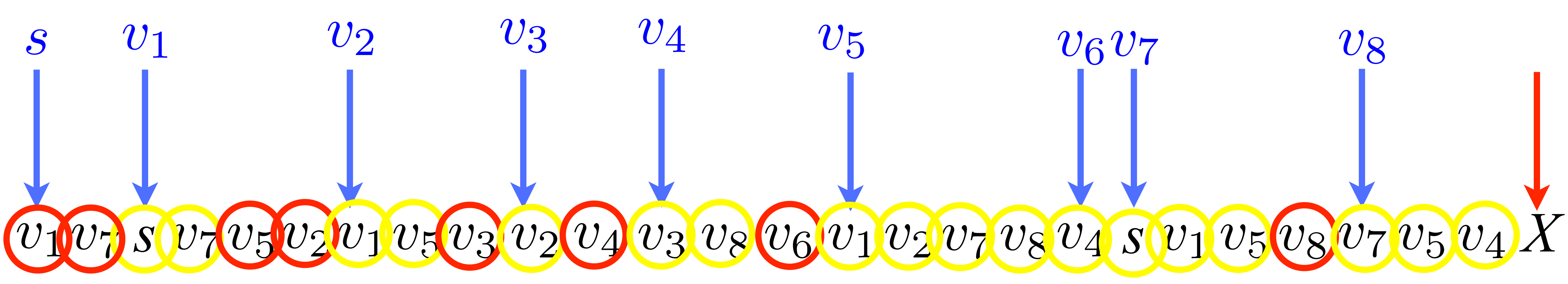
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle v∈ R
    - 2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        - 2.2.1. R:=R\{v}
    - 2.3. ELSE {
        - 2.3.1. Wähle ein w∈ V\Y mit e={v,w} ∈ E
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





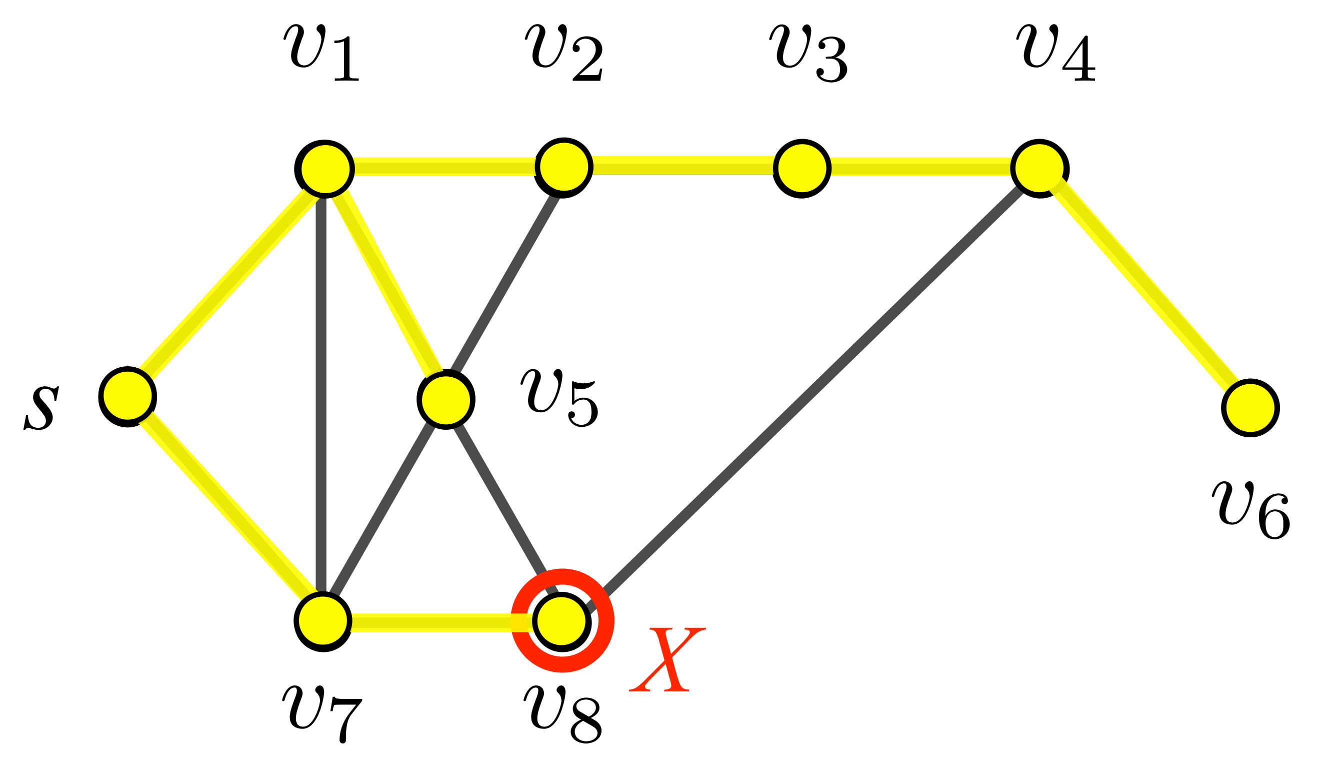
Algorithmus 3.7

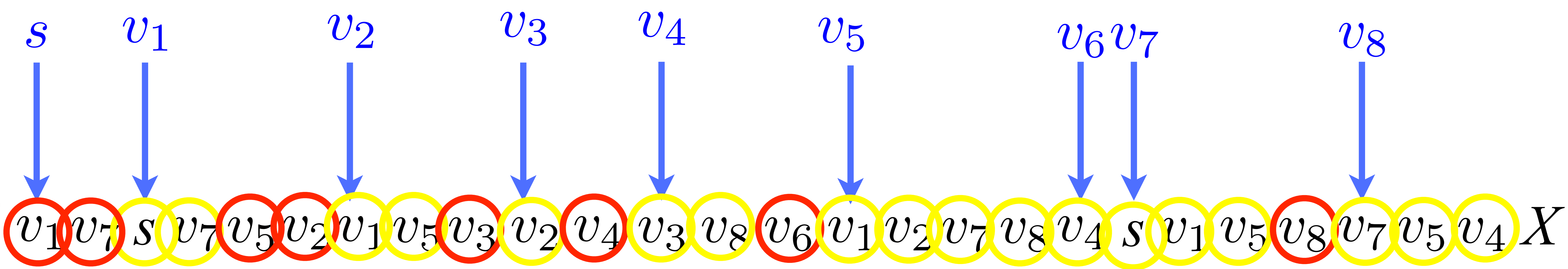
INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈ R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w} ∈ E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

     }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

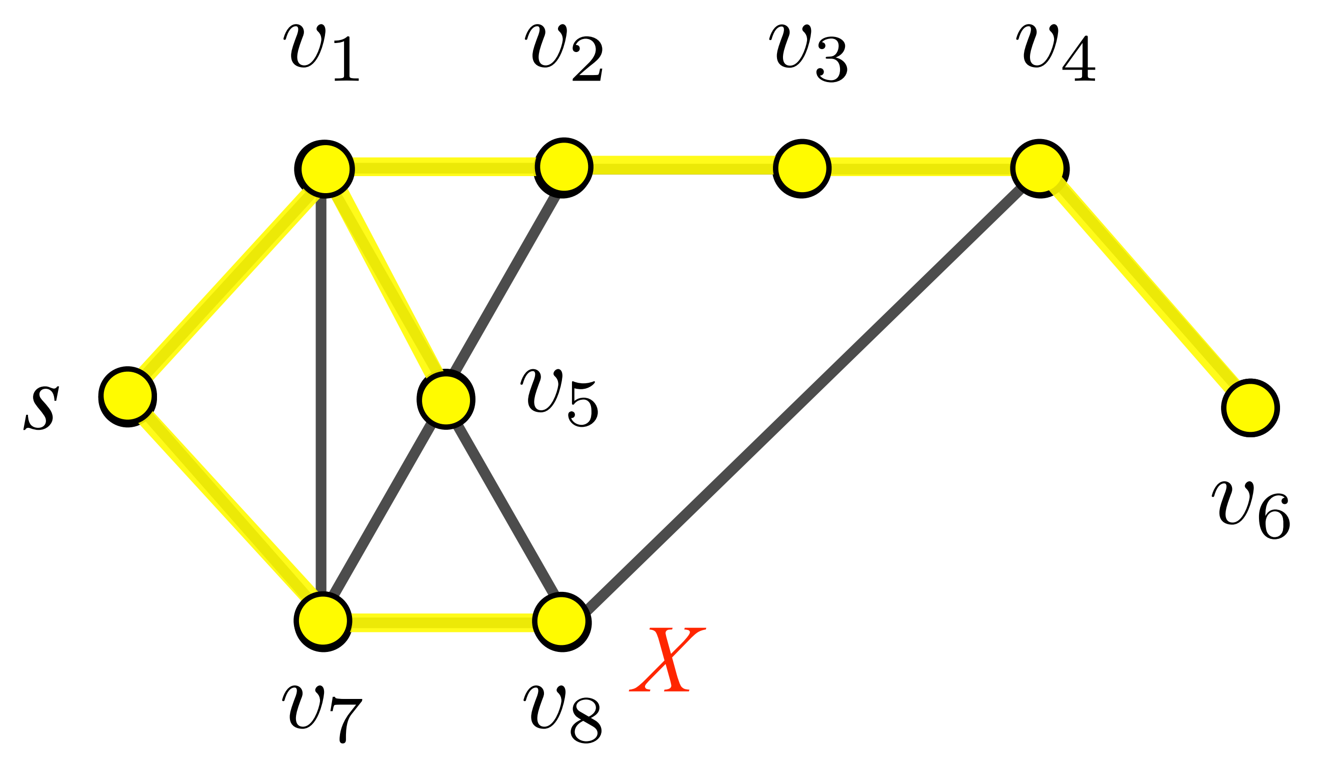
OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

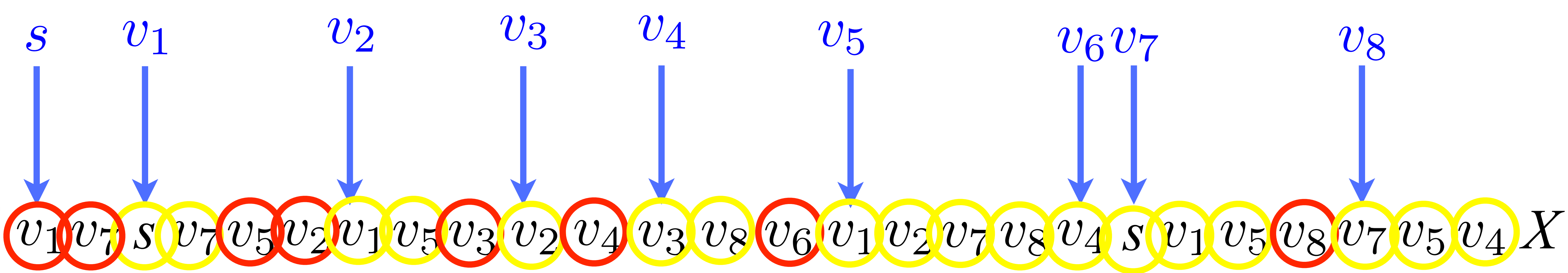
Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle v∈R
   - 2.2. IF (es gibt kein w∈V\Y mit e={v,w}∈E) THEN
     - 2.2.1. R:=R\{v}
   - 2.3. ELSE {
     - 2.3.1. Wähle ein w∈V\Y mit e={v,w}∈E
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

       }

   }
3. STOP





Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

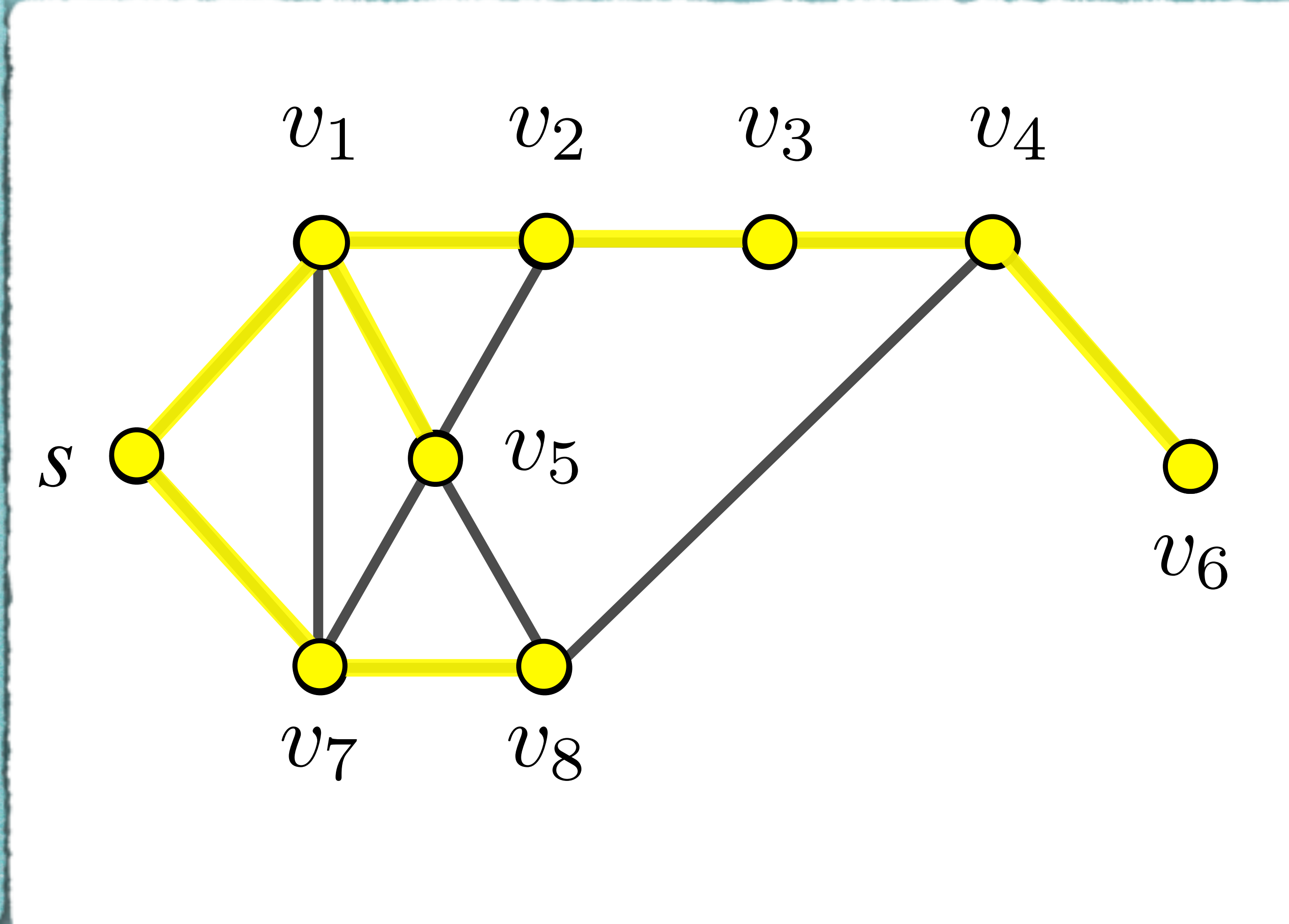
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

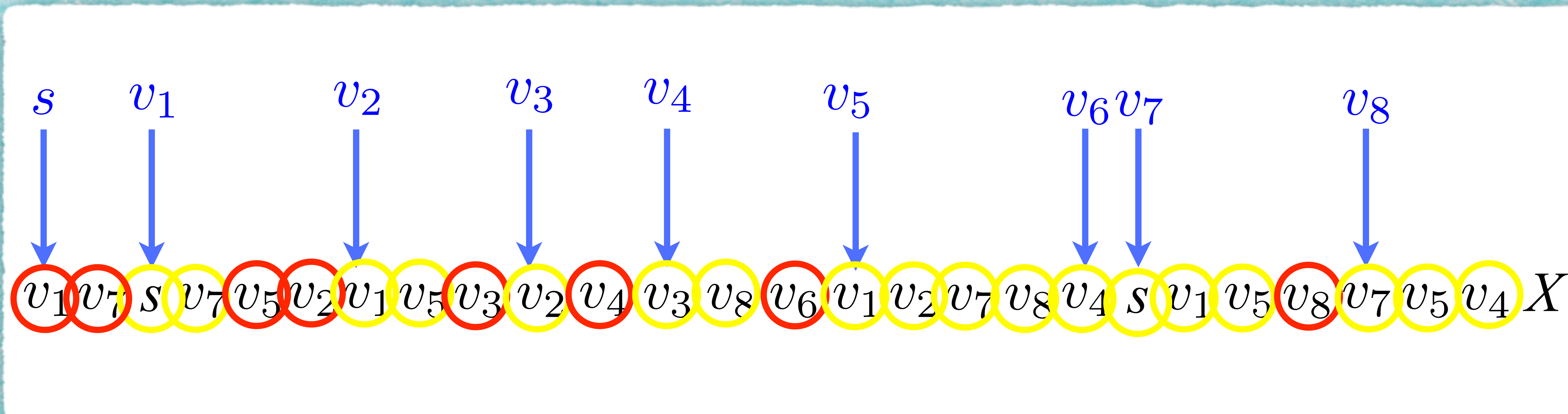
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    - 2.1. Wähle $v \in R$
    - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
        - 2.2.1. $R:=R \setminus \{v\}$
    - 2.3. ELSE {
        - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
        - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

    }
3. STOP





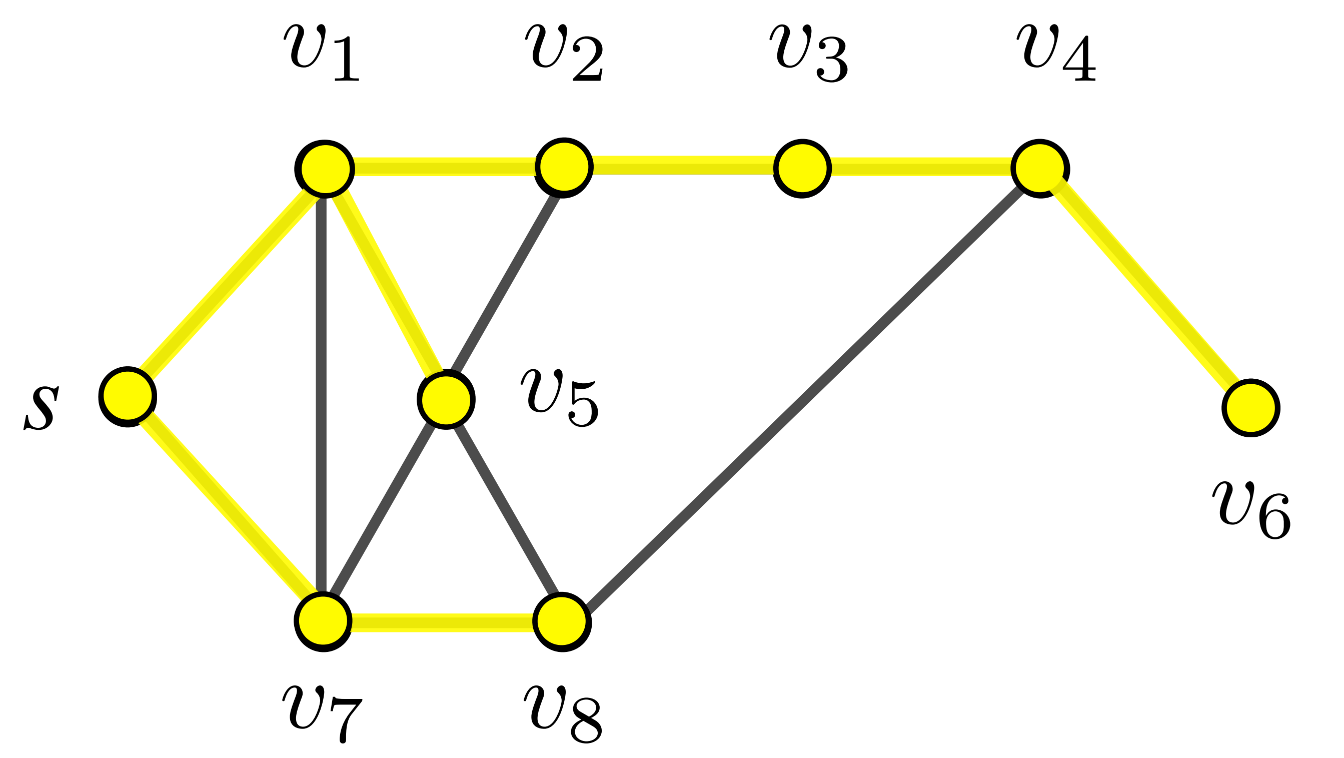
Algorithmus 3.7

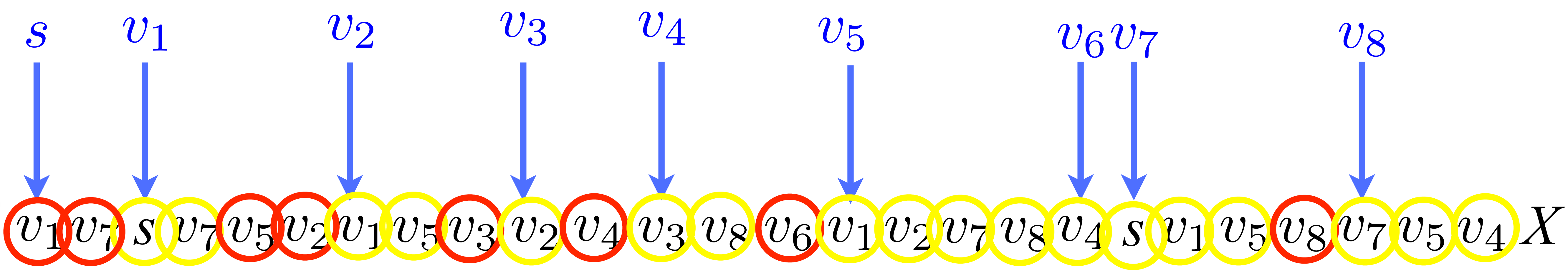
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge Y ⊆ V, die von s aus erreichbar ist,

Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
    2.1. Wähle v∈ R
    2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
        2.2.1. R:=R\{v}
    2.3. ELSE {
        2.3.1.Wähle ein w ∈V\Y mit e={v,w} ∈ E
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
        }
    }
3. STOP





Satz 3.13

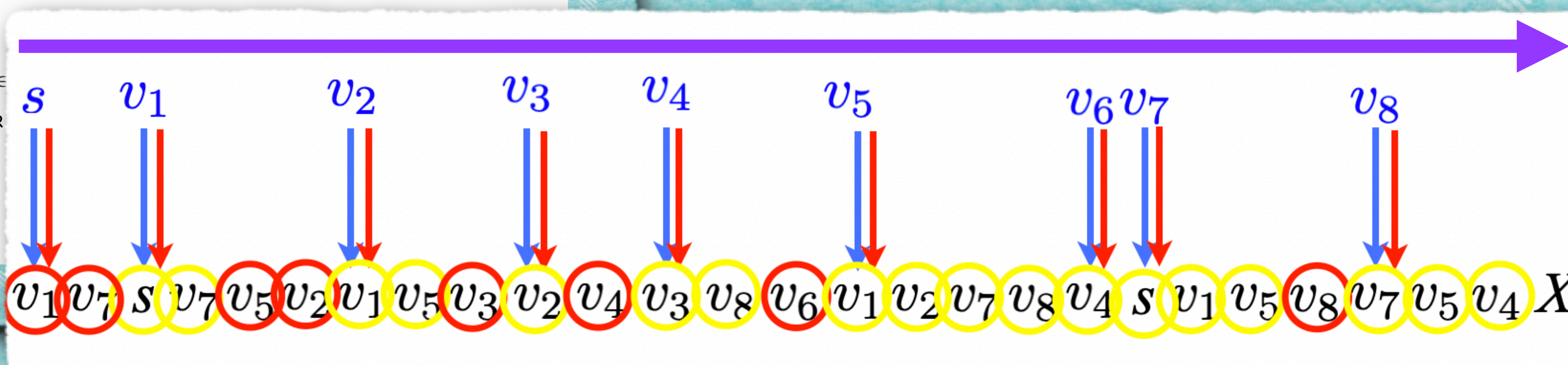
Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

Algorithmus 3.7

INPUT:	Graph G = (V,E), Knoten s
OUTPUT:	Knotenmenge Y ⊆ V, die von s aus erreichbar ist,
	Kantenmenge T ⊆ E, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   2.1. Wähle v∈ R
   2.2. IF (es gibt kein w∈ V\Y mit e={v,w}∈E) THEN
      2.2.1. R:=R\{v}
   2.3. ELSE {
      2.3.1.Wähle ein w∈
      2.3.2. Setze R := R
      }
   }
3. STOP

Adjazenzliste!



Demnächst mehr!

s.fekete@tu-bs.de