



Kapitel 3: Suche in Graphen

Algorithmen und Datenstrukturen
WS 2026/27

Prof. Dr. Sándor Fekete

A & D

I get the job done. What the hell do you want?

Can you make it without killing yourself?



Algorithmus

Datenstruktur

Algorithmus 3.7

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

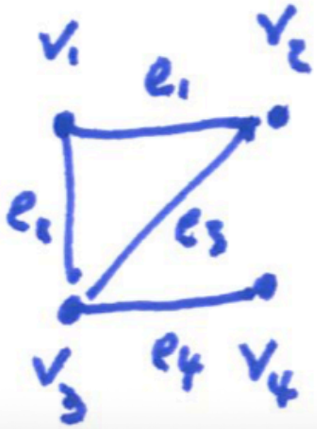
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

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

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

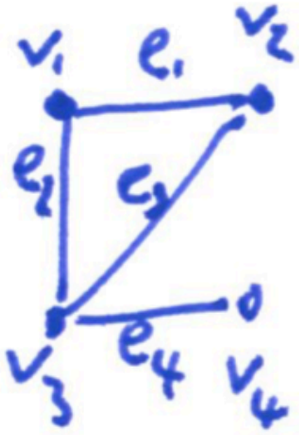
(1) Inzidenzmatrix



	e_1	e_2	e_3	e_4
v_1	1	1	0	0
v_2	1	0	1	0
v_3	0	1	1	1
v_4	0	0	0	1

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

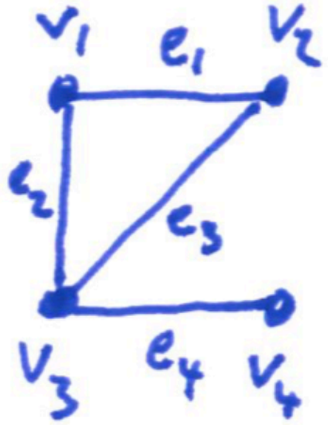
(2) Adjazenzmatrix



	v_1	v_2	v_3	v_4
v_1	0	1	1	0
v_2	1	0	1	0
v_3	1	1	0	1
v_4	0	0	1	0

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

(3) Kantenliste



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

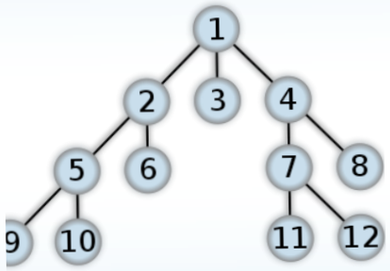
Benötigt wird eine Kantennummerierung!

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

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

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

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



Kapitel 3.7: Wachstum von Funktionen

Algorithmen und Datenstrukturen
WS 2025/26

Prof. Dr. Sándor Fekete

Idee: Verhalten von Funktionen

asymptotisch abschätzen und vereinfachen

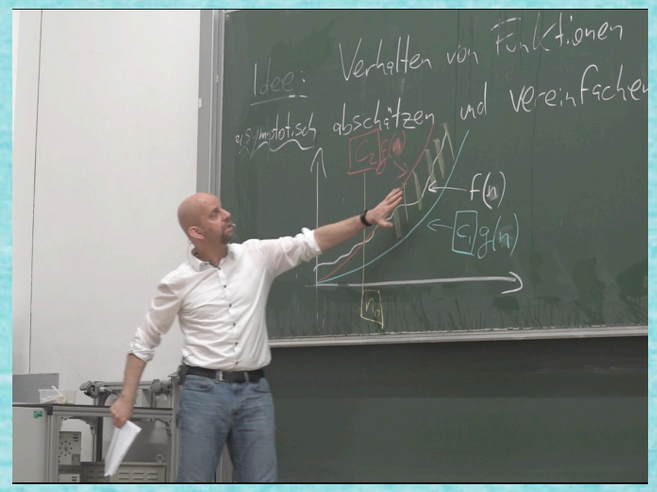
$c_2 g(n)$

$f(n)$

$c_1 g(n)$

n_0

Wofür ist das gut?



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

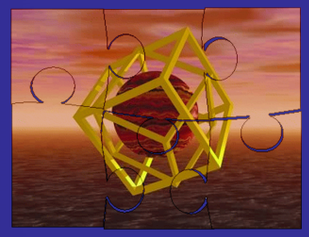
$$\Theta(m\log n)$$

Einfach so: $O(n^2)$

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

Ein algorithmisches Problem

Gegeben: n Puzzleteile

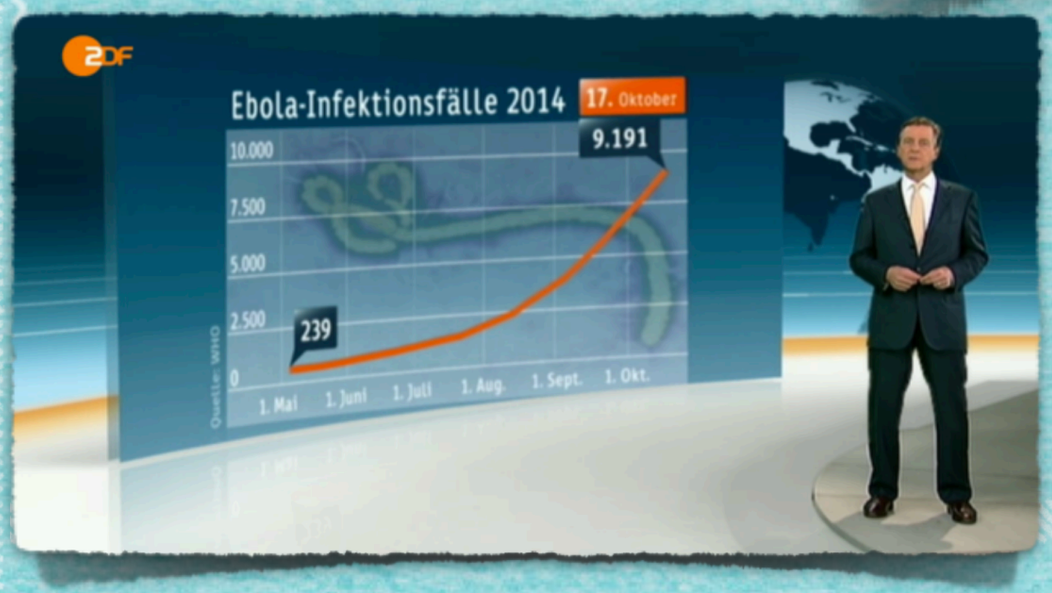


Einfach so: $O(n^2)$

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

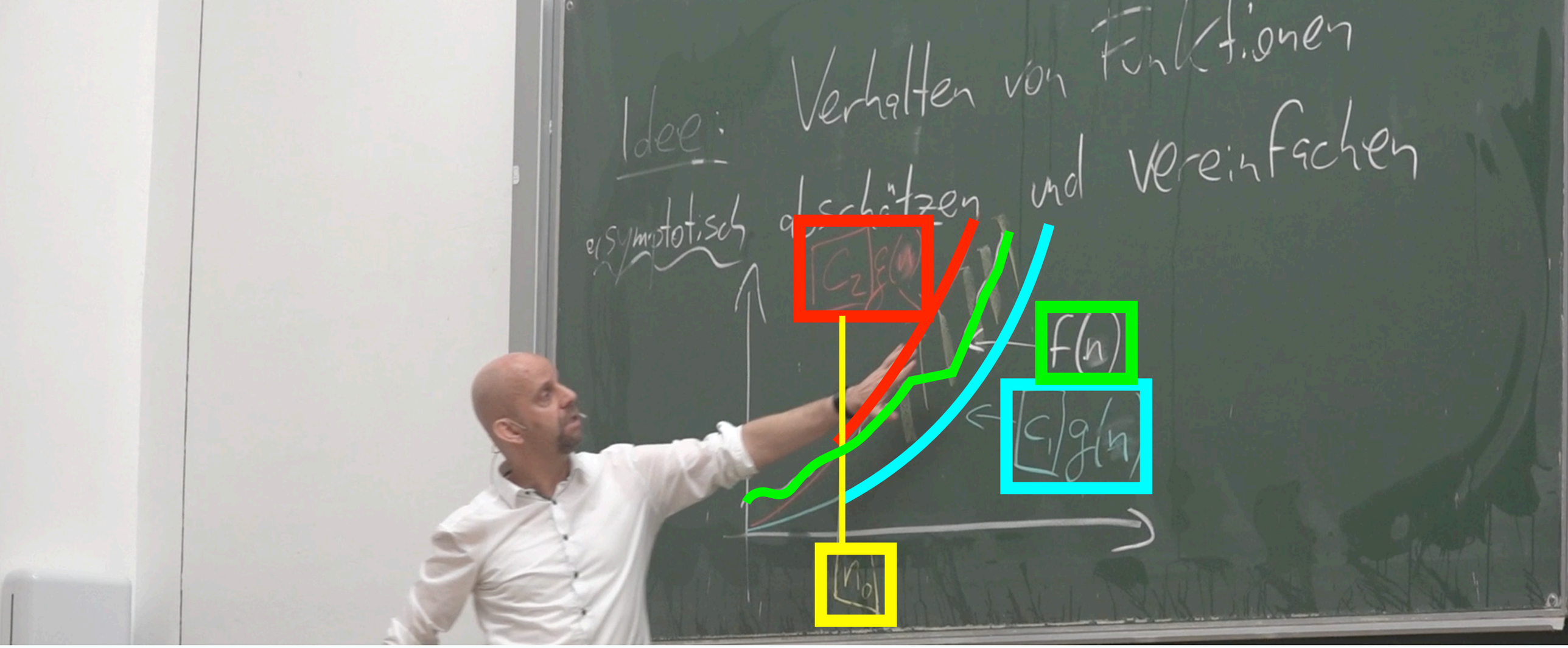
Raffiniert sortiert: $O(n\log n)$

Gesucht: Eine systematische Methode zum Puzzeln



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

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



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

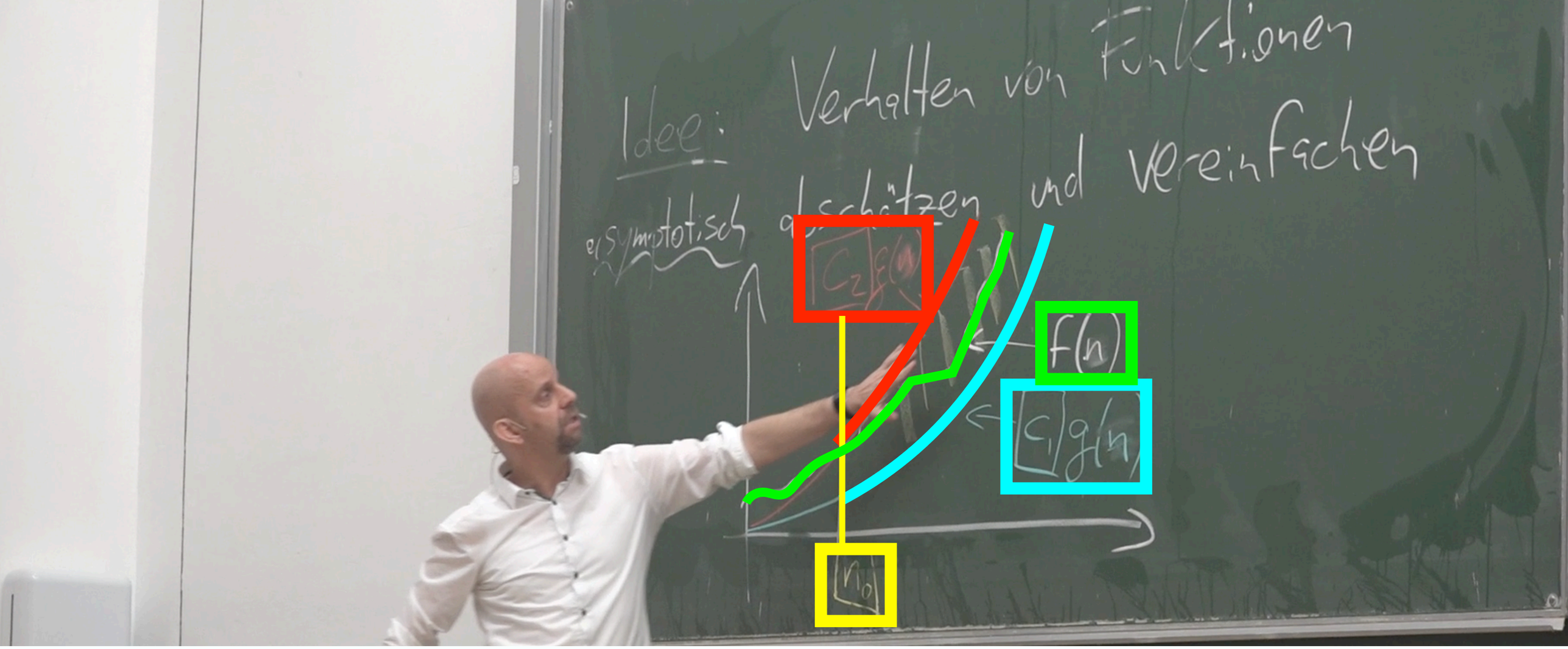
$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ für alle $n \geq n_0$

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

Quiz!

s.fekete@tu-bs.de

Idee: Verhalten von Funktionen asymptotisch abschätzen und vereinfachen



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

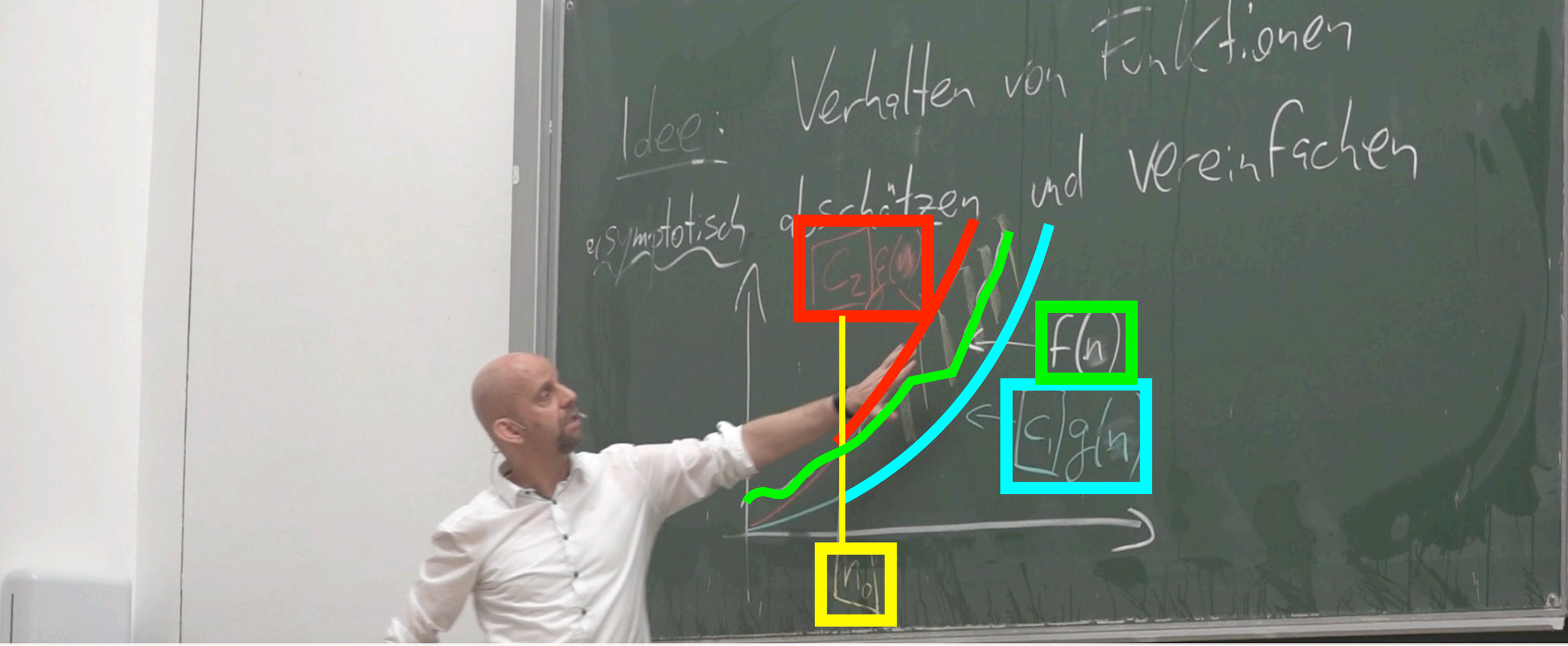
$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ für alle $n \geq n_0$

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

Idee: Verhalten von Funktionen asymptotisch abschätzen und vereinfachen

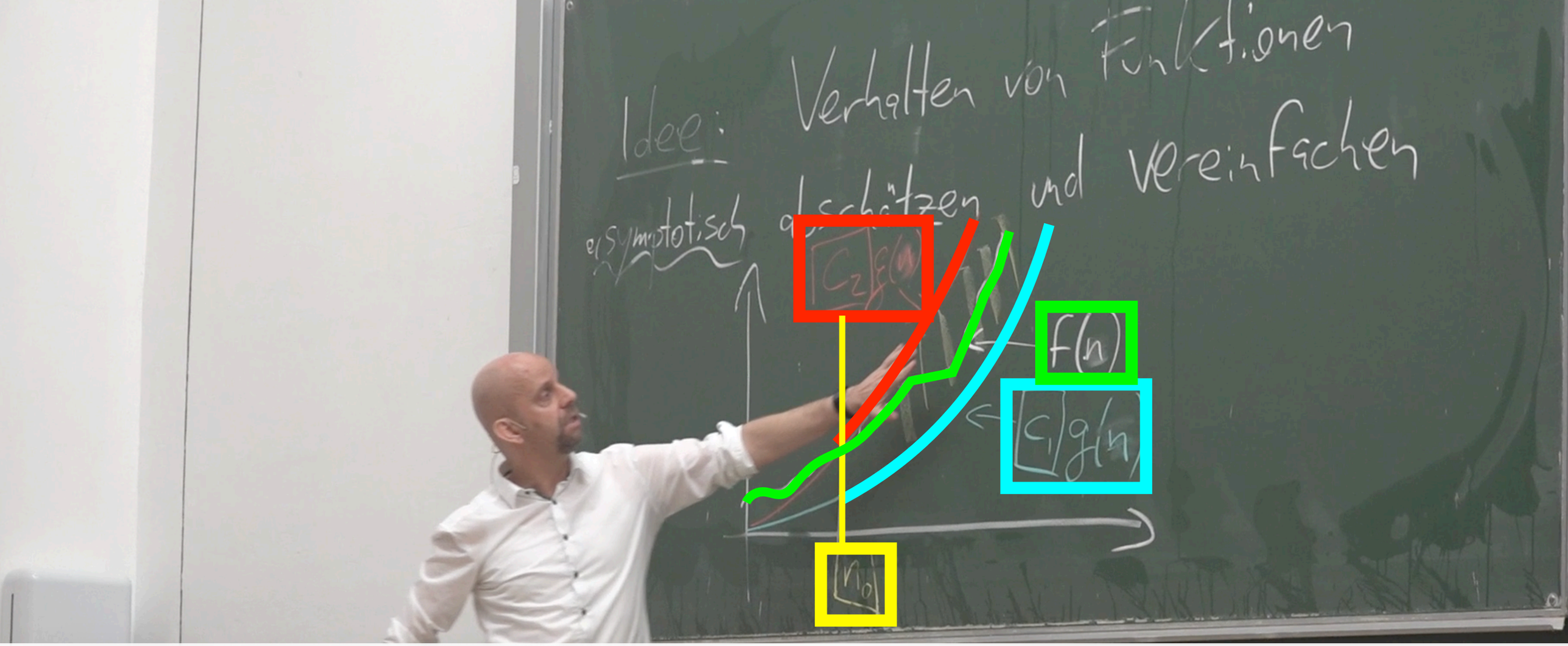
$c_2 f(n)$ $f(n)$ $c_1 g(n)$ n_0

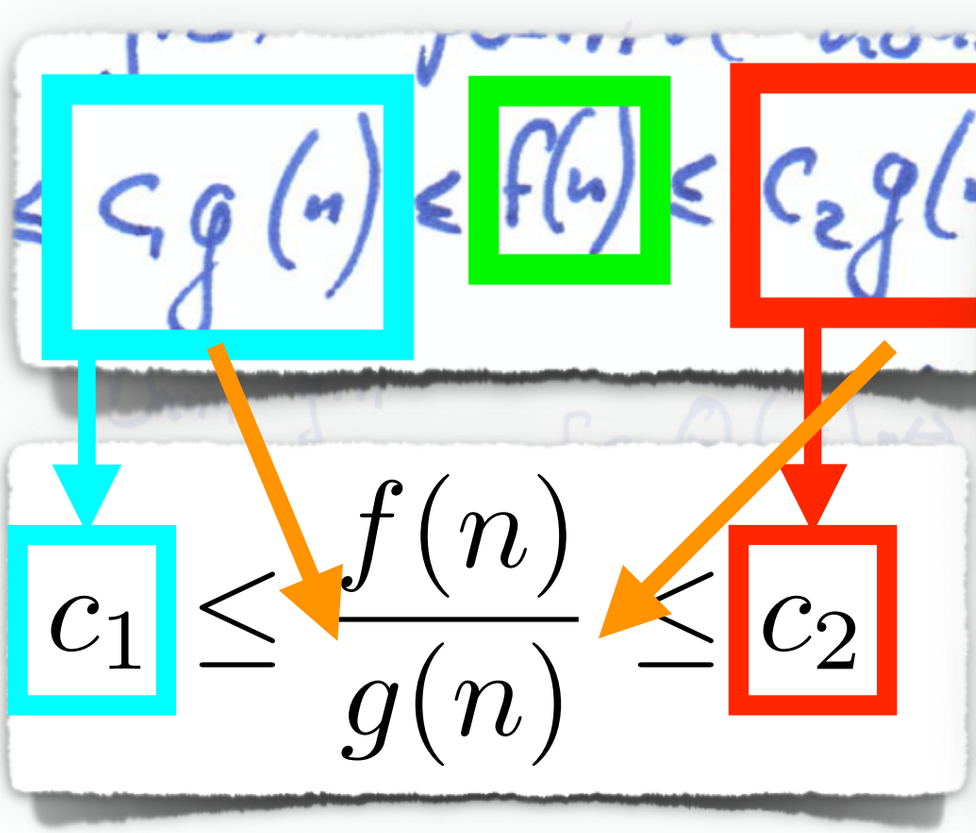
$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$



$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

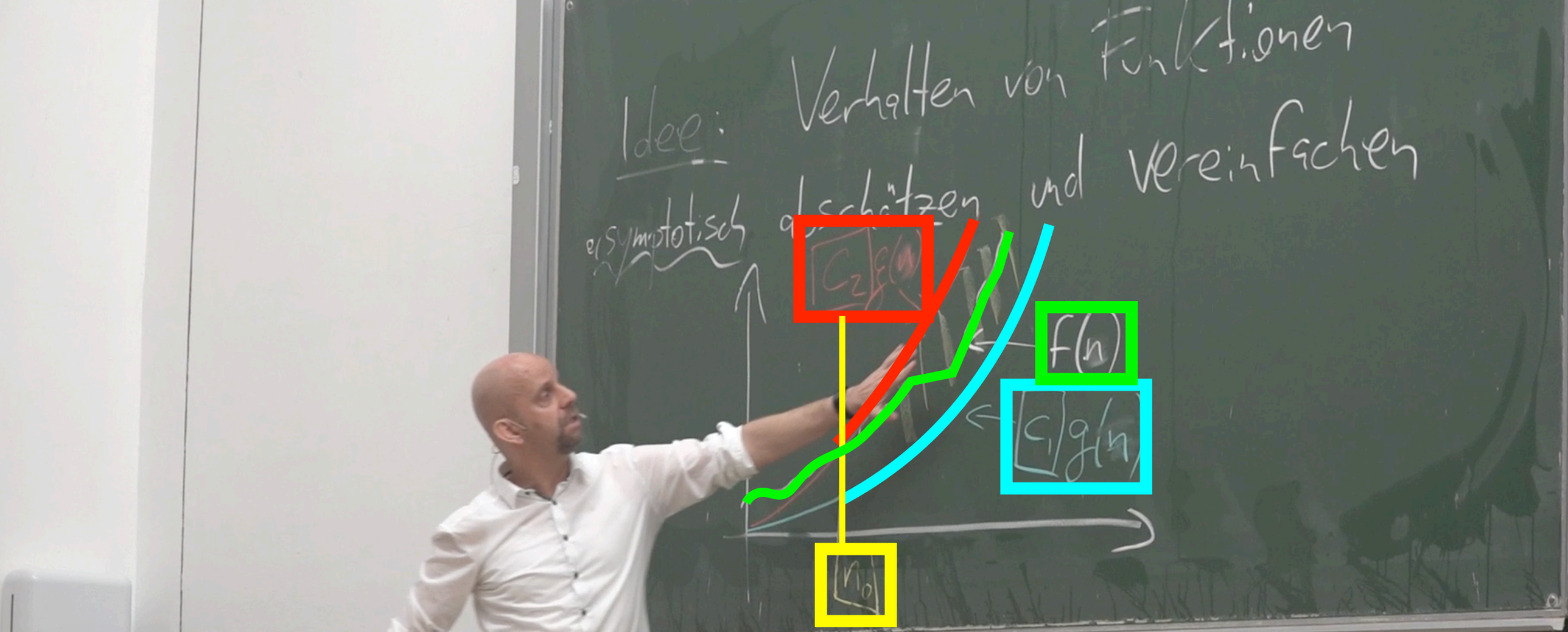




Frage 7:

Die Funktion $f(n) := 5n^6 + 4n^4 + 7n^3 + 27n - 9$ liegt...

- ... nur in $O(n^6)$.
- ... nur in $\Omega(n^6)$.
- ... in $\Theta(n^6)$.



$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

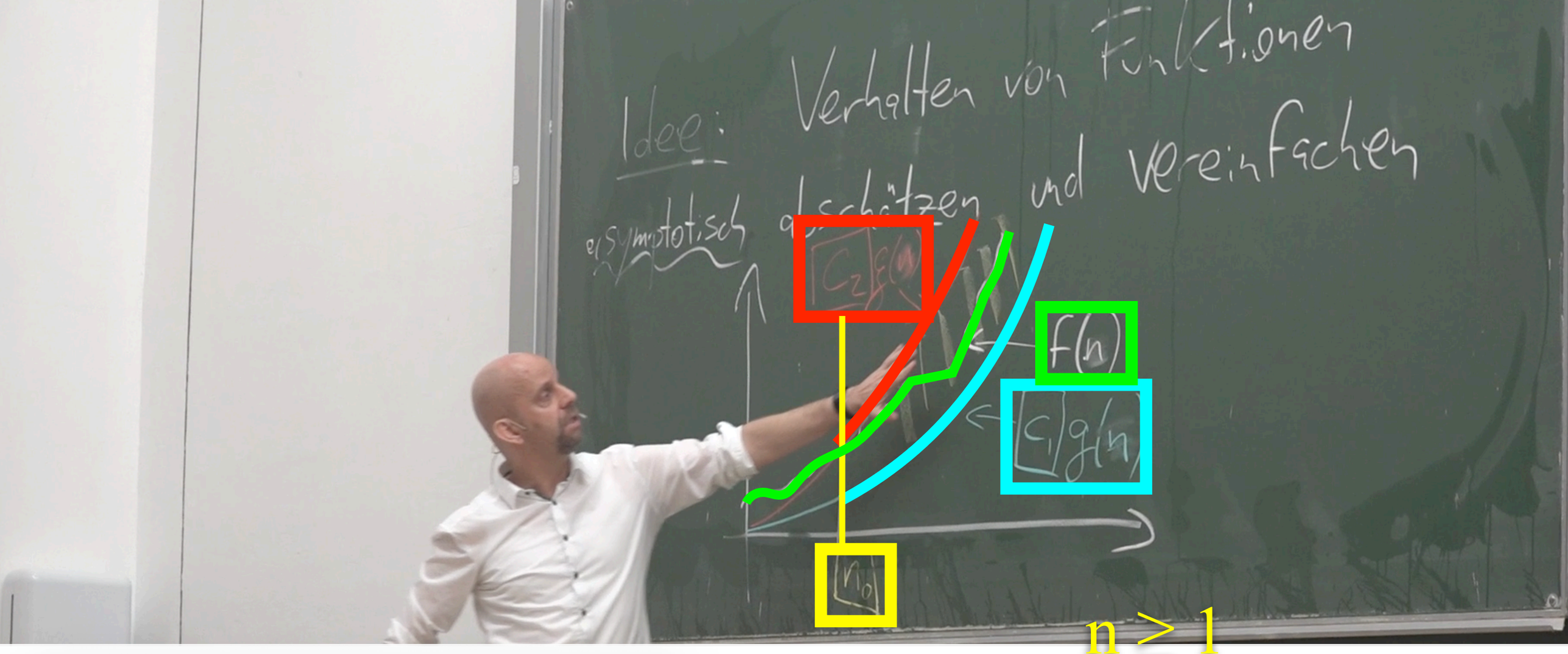
$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

$$\frac{f(n)}{g(n)} = 5 + \frac{4}{n^2} + \frac{7}{n^3} + \frac{27}{n^5} - \frac{9}{n^6}$$

Frage 7:

Die Funktion $f(n) := 5n^6 + 4n^4 + 7n^3 + 27n - 9$ liegt...

- ... nur in $O(n^6)$.
- ... nur in $\Omega(n^6)$.
- ... in $\Theta(n^6)$.



$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

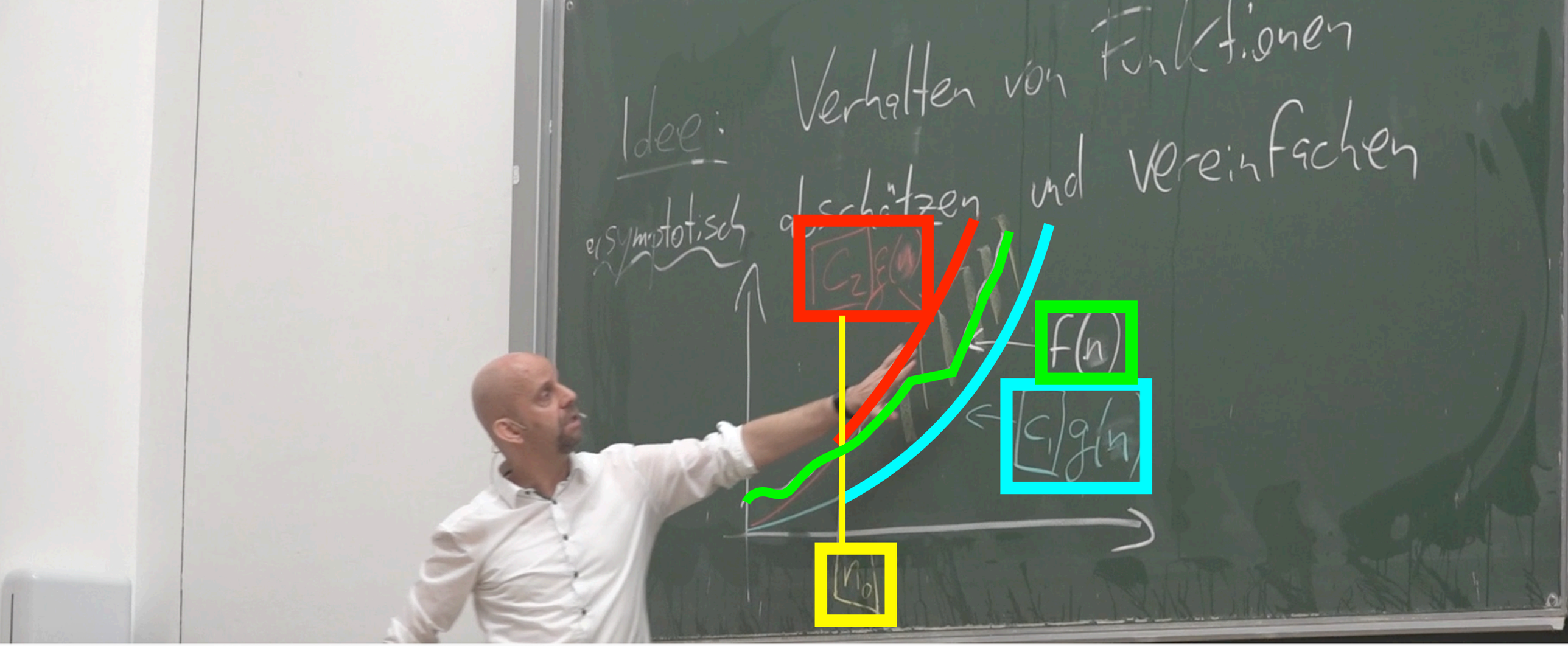
$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

$$5 \leq \frac{f(n)}{g(n)} = 5 + \frac{4}{n^2} + \frac{7}{n^3} + \frac{27}{n^5} - \frac{9}{n^6} \leq 43$$

Frage 7:

Die Funktion $f(n) := 5n^6 + 4n^4 + 7n^3 + 27n - 9$ liegt...

- ... nur in $O(n^6)$.
- ... nur in $\Omega(n^6)$.
- ... in $\Theta(n^6)$. ✓

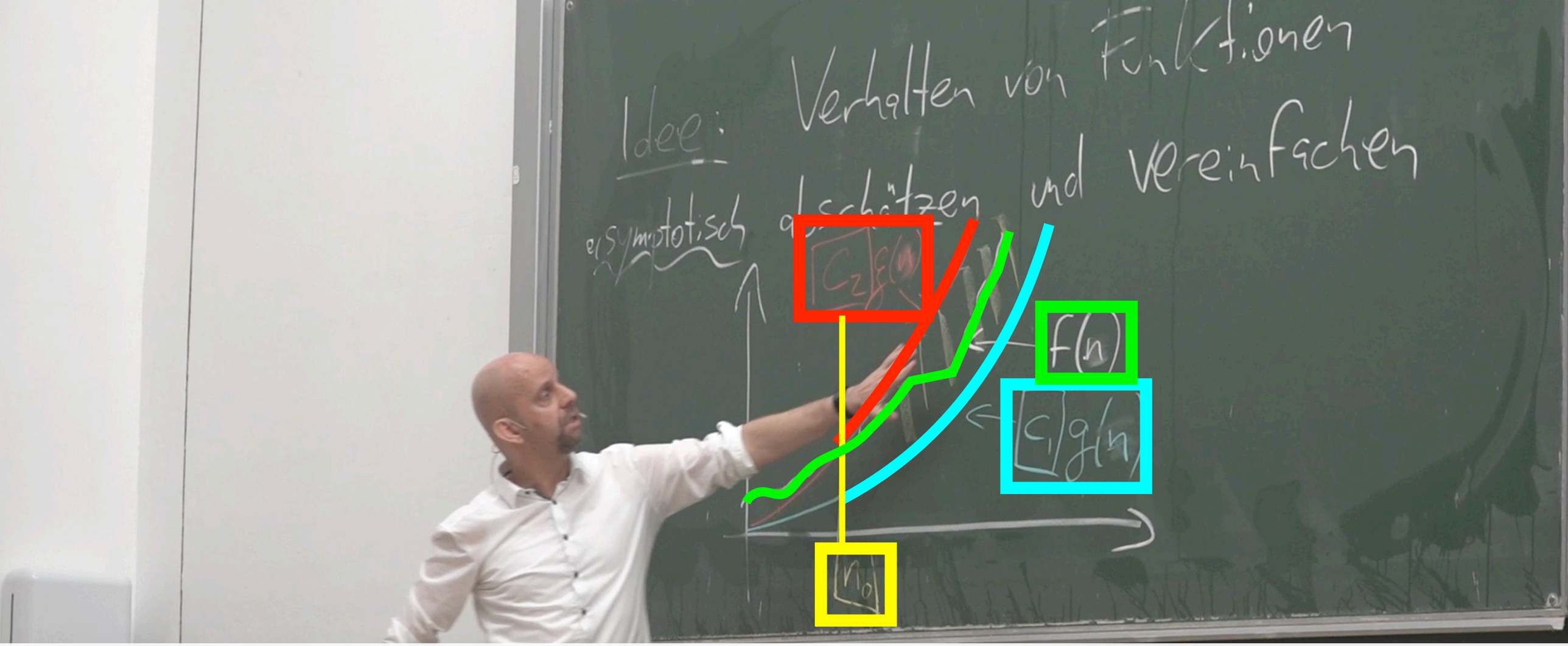


$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

Frage 8:

Die Funktion $f(n) := 2^n$ liegt...

- ... nur in $O(3^n)$.
- ... nur in $\Omega(3^n)$.
- ... in $\Theta(3^n)$.



$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

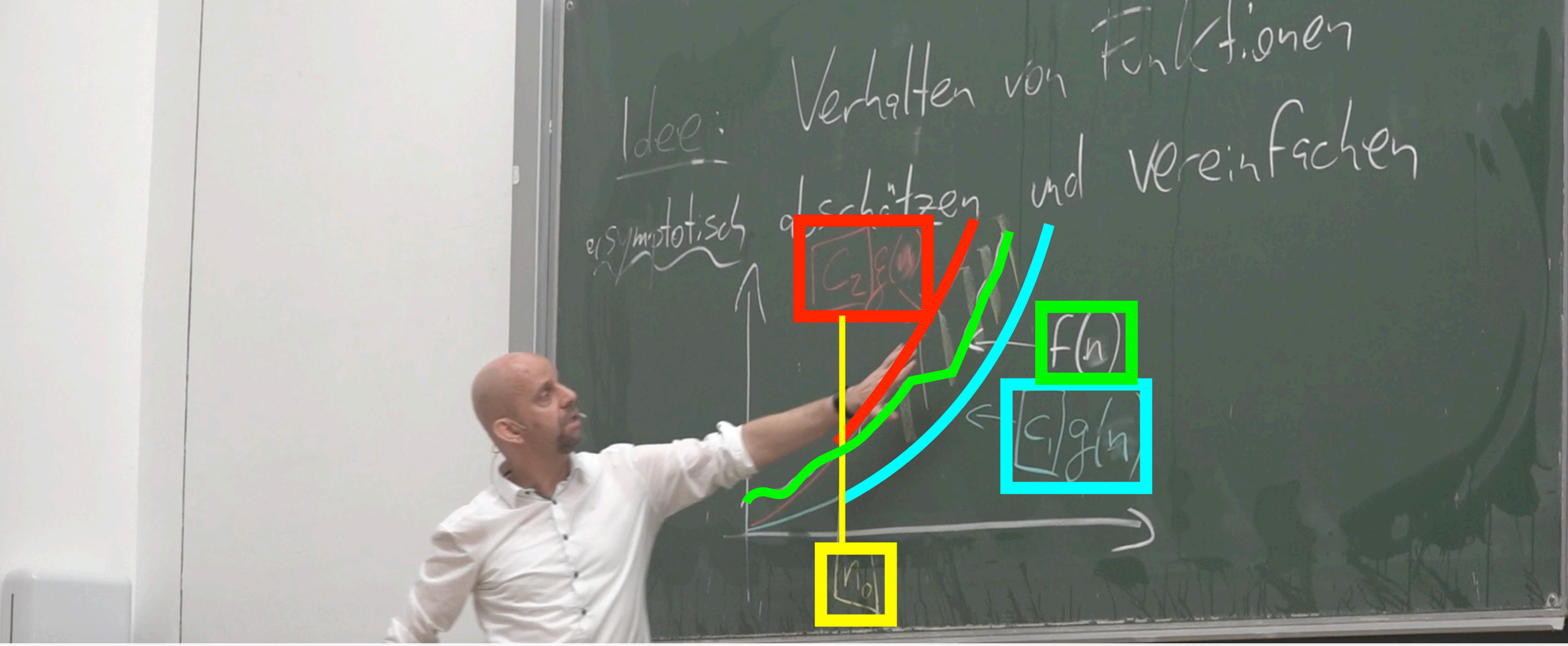
$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

Frage 8:

Die Funktion $f(n) := 2^n$ liegt...

- ... nur in $O(3^n)$.
- ... nur in $\Omega(3^n)$.
- ... in $\Theta(3^n)$.

$$\frac{f(n)}{g(n)} = \frac{2^n}{3^n} = \left(\frac{2}{3}\right)^n$$



$$c_1 \leq \frac{f(n)}{g(n)} \leq c_2$$

Frage 8:

Die Funktion $f(n) := 2^n$ liegt...

- ... nur in $O(3^n)$. ✓
- ... nur in $\Omega(3^n)$.
- ... in $\Theta(3^n)$.

$$\frac{f(n)}{g(n)} = \frac{2^n}{3^n} = \left(\frac{2}{3}\right)^n$$

SATZ 3.12

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

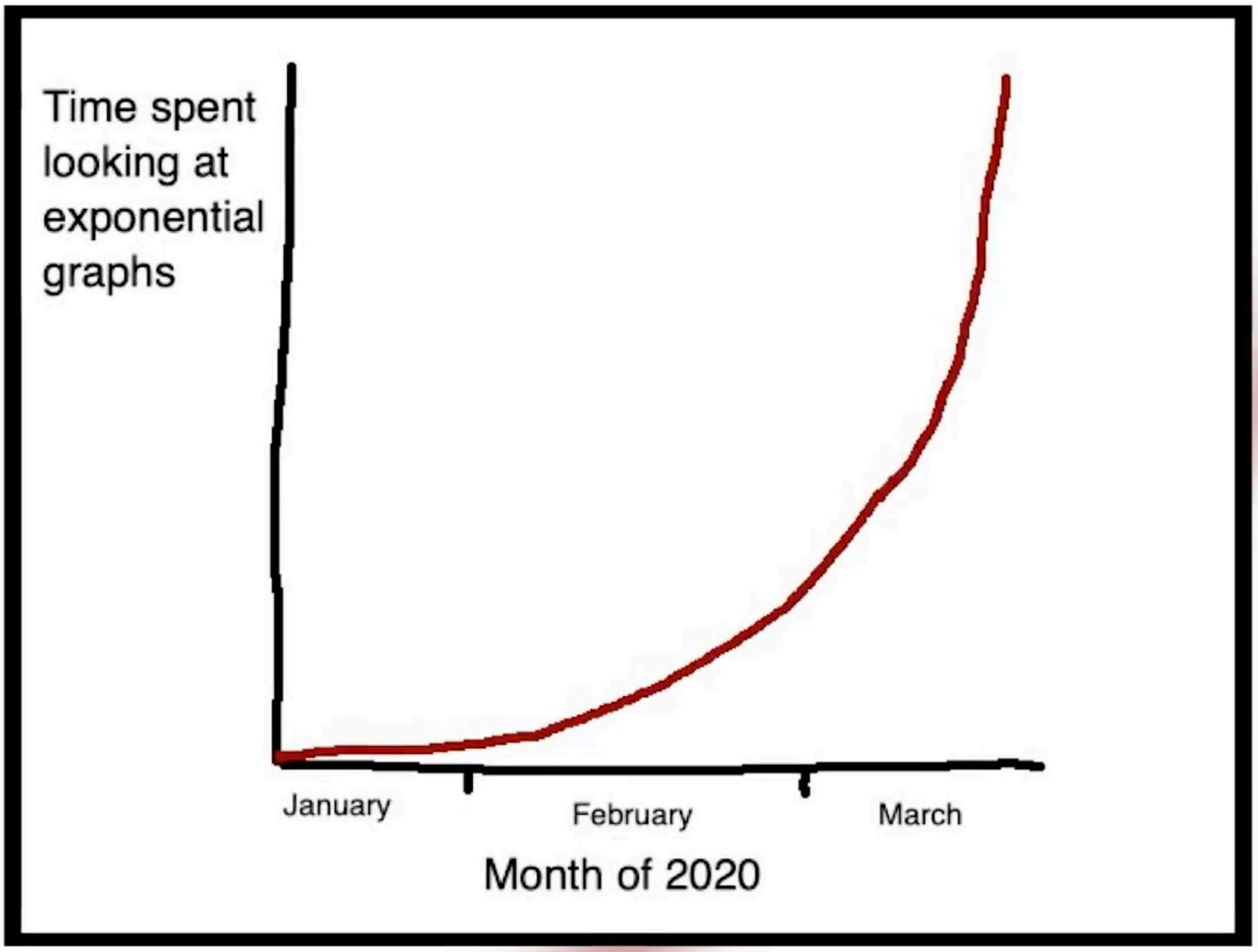
Dann gilt

(1) $f \in \Theta(g) \iff g \in \Theta(f)$

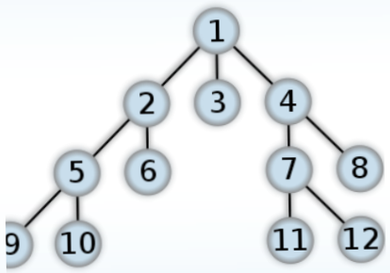
(2) $f \in \Theta(g) \iff f \in O(g)$ und $f \in \Omega(g)$.

(3) $f \in O(g) \iff g \in \Omega(f)$

Beweis: Übung!


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





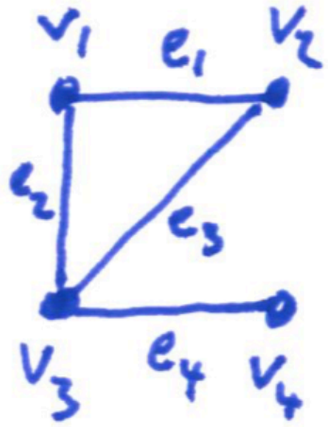


Kapitel 3.6: Datenstrukturen für Graphen

Algorithmen und Datenstrukturen
WS 2026/27

Prof. Dr. Sándor Fekete

(3) Kantenliste



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

Benötigt wird eine Kantennummerierung!



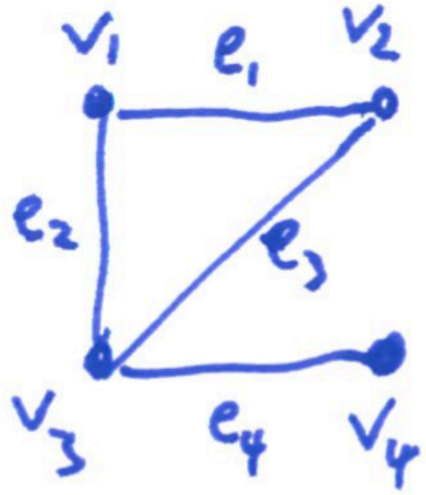
xkcd

A WEBCOMIC OF ROMANCE, SARCASM, MATH, AND LANGUAGE.



HOME ORGANIZATION TIP:
JUST GIVE UP.

(4) Adjazenzliste



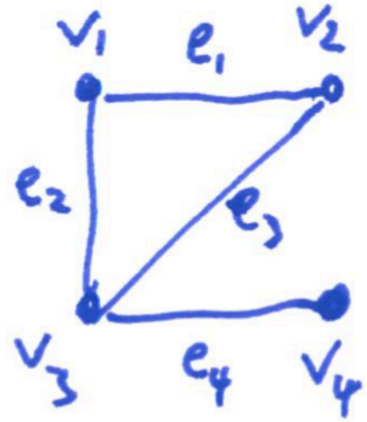
V_1 : V_2, V_3 ;

V_2 : V_1, V_3 ;

V_3 : V_1, V_2, V_4 ;

V_4 : V_3 ;

(4) Adjazenzliste

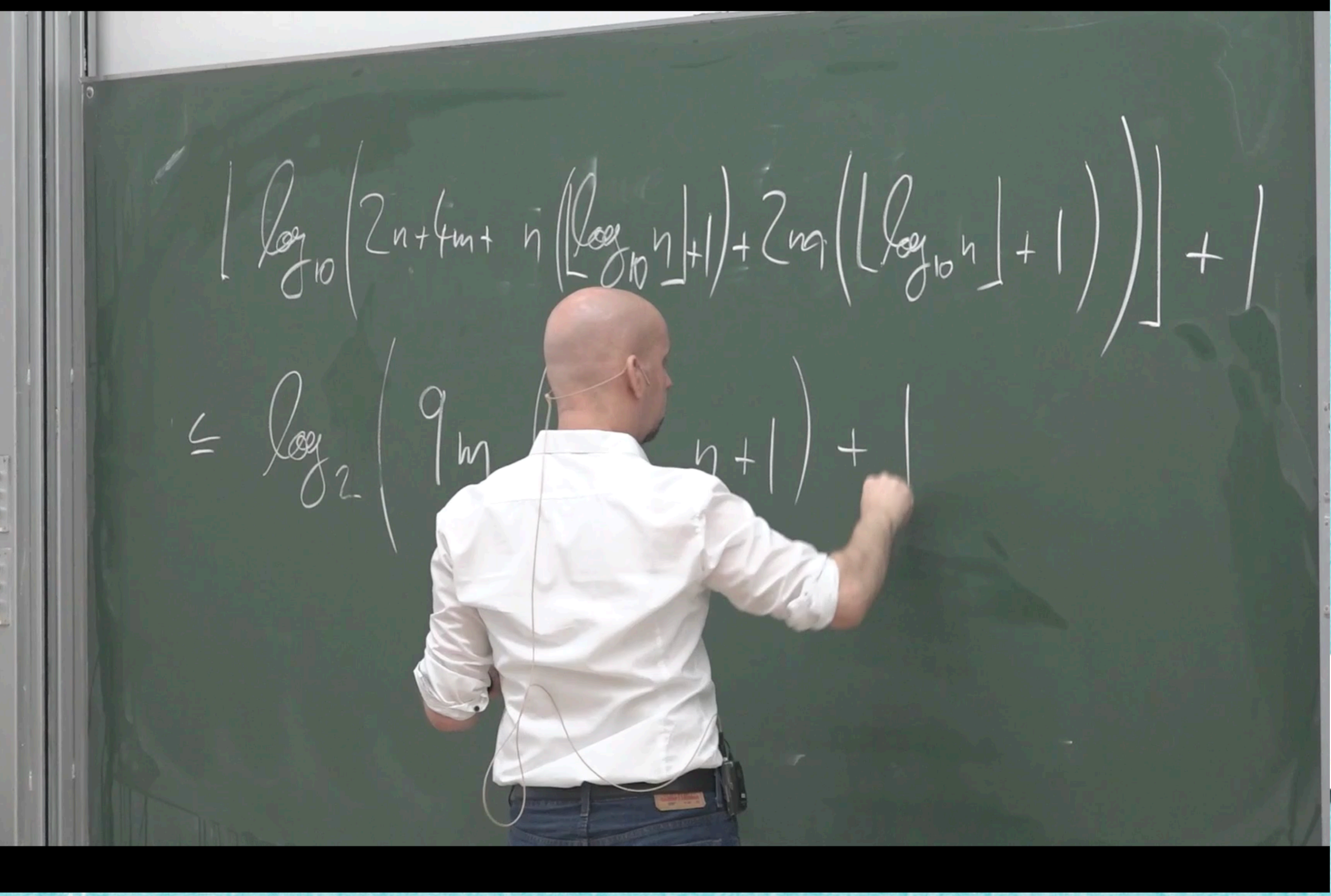


v_1 : v_2, v_3 ;
v_2 : v_1, v_3 ;
v_3 : v_1, v_2, v_4 ;
v_4 : v_3 ;

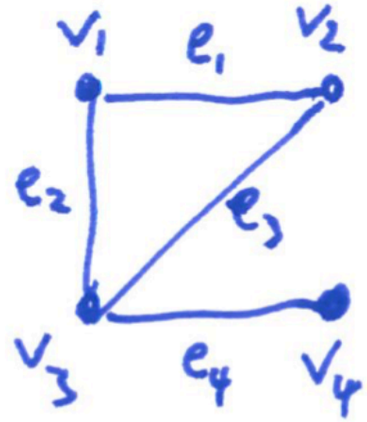
↓ ↓ ↓ ↓

v_2, v_3 ; v_1, v_3 ; v_1, v_2, v_4 ; v_3

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


$\lfloor \log_{10}\left(2n+4m+n\left(\lfloor \log_{10} n \rfloor+1\right)+2n\left(\lfloor \log_{10} n \rfloor+1\right)\right)\rfloor+1$
$\leq \log_2\left(9m\left(\quad n+1\right)\right)+1$
$]]+1$

(4) Adjazenzliste



v_1 : v_2, v_3 ;

v_2 : v_1, v_3 ;

v_3 : v_1, v_2, v_4 ;

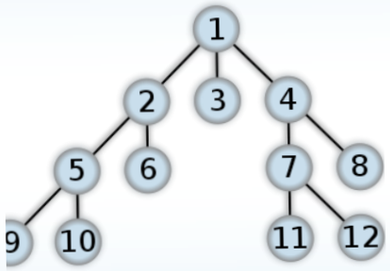
v_4 : v_3 ;

↓ ↓ ↓ ↓

v_2, v_3 ; v_1, v_3 ; v_1, v_2, v_4 ; v_3

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

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



Kapitel 3.8: Laufzeit von DFS und BFS

Algorithmen und Datenstrukturen
WS 2025/26

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

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 {

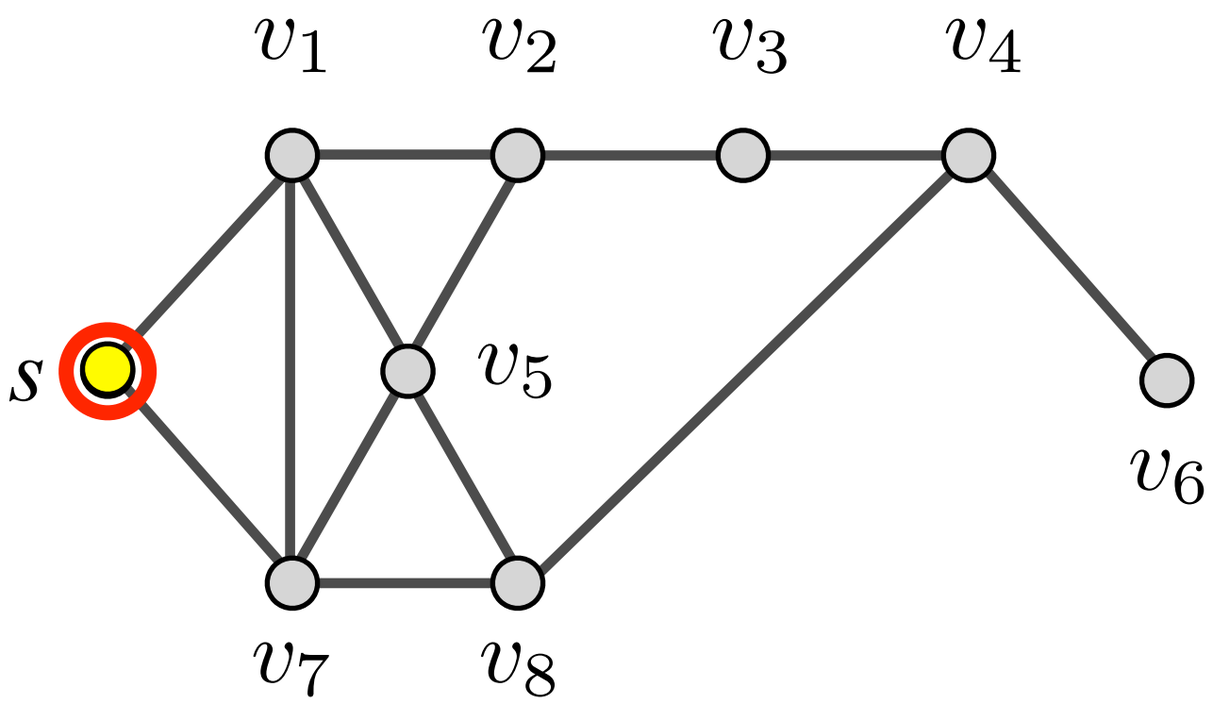
        2.3.1.Wähle ein w∈V\Y mit e={v,w} ∈ E

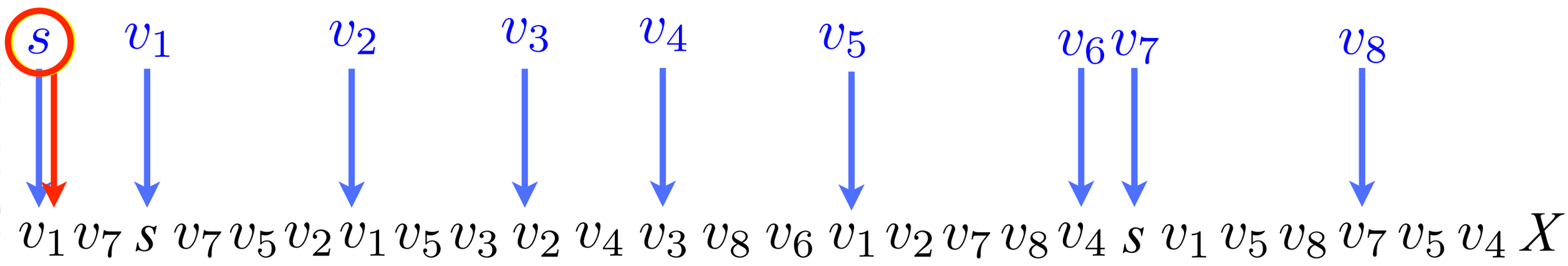
        2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}

        }

}

3. STOP





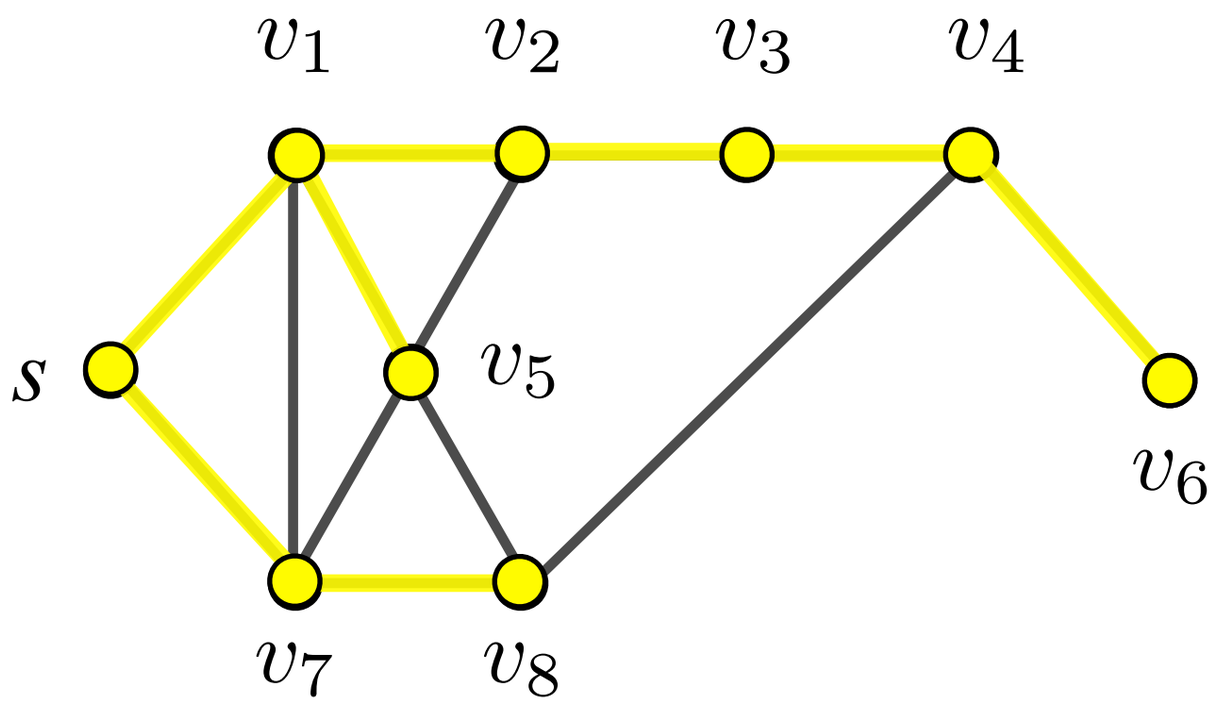
Algorithmus 3.7

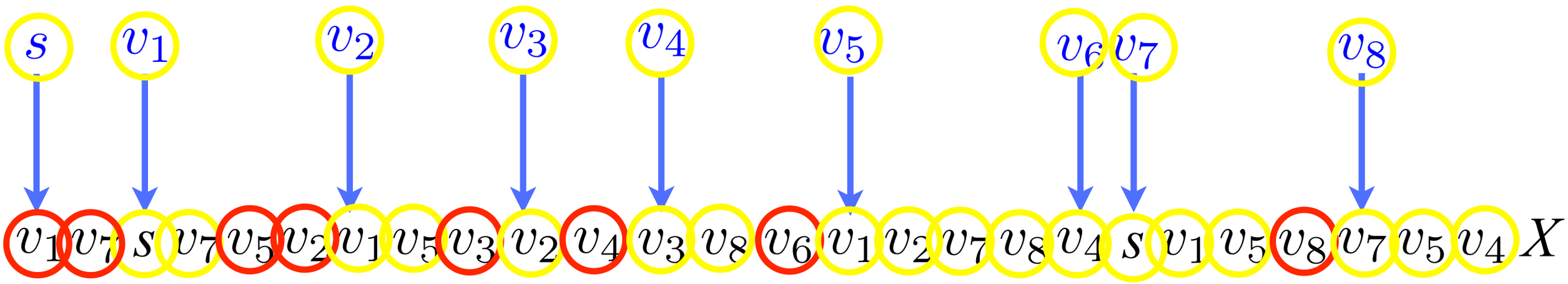
INPUT: Graph G = (V,E), Knoten s

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





Satz 3.13

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

Algorithmus 3.7

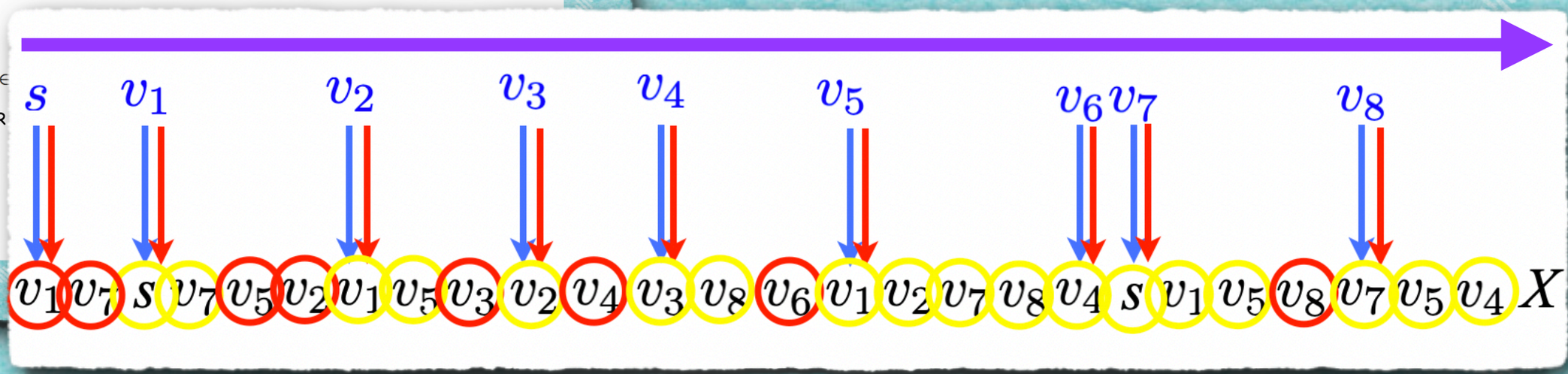
INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, 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∈
     - 2.3.2. Setze R := R
     - }
   - }
3. STOP

Adjazenzliste!



Demnächst mehr!

s.fekete@tu-bs.de