

Kapitel 3.3: Zusammenhangskomponenten

*Algorithmen und Datenstrukturen
WS 2021/22*

Prof. Dr. Sándor Fekete

DEFINITION 3.6 (Wald, Baum)

- (1) Ein Wald ist ein kreisfreier Graph.
- (2) Ein Baum ist eine Zusammenhangskomponente in einem Wald.
(Also: ein kreisfreier, zusammenhängender Graph)
- (3) Ein aufspannender Baum ist ein Baum, der alle Knoten verbindet. (Manchmal auch: Spannbaum. Englisch: "spanning tree")

Algorithmus 3.7

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

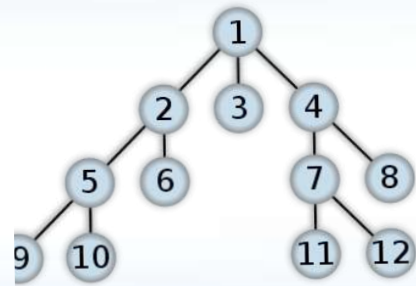
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

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

Satz 3.8. *Der Algorithmus 3.7 ist*

(1) endlich

(2) korrekt.



Kapitel 3.4: *Warteschlange und Stapel*

Algorithmen und Datenstrukturen *WS 2021/22*

Prof. Dr. Sándor Fekete

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

Wie verwalten wir R?

R



Wähle v!

Waschkorb

A & D

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

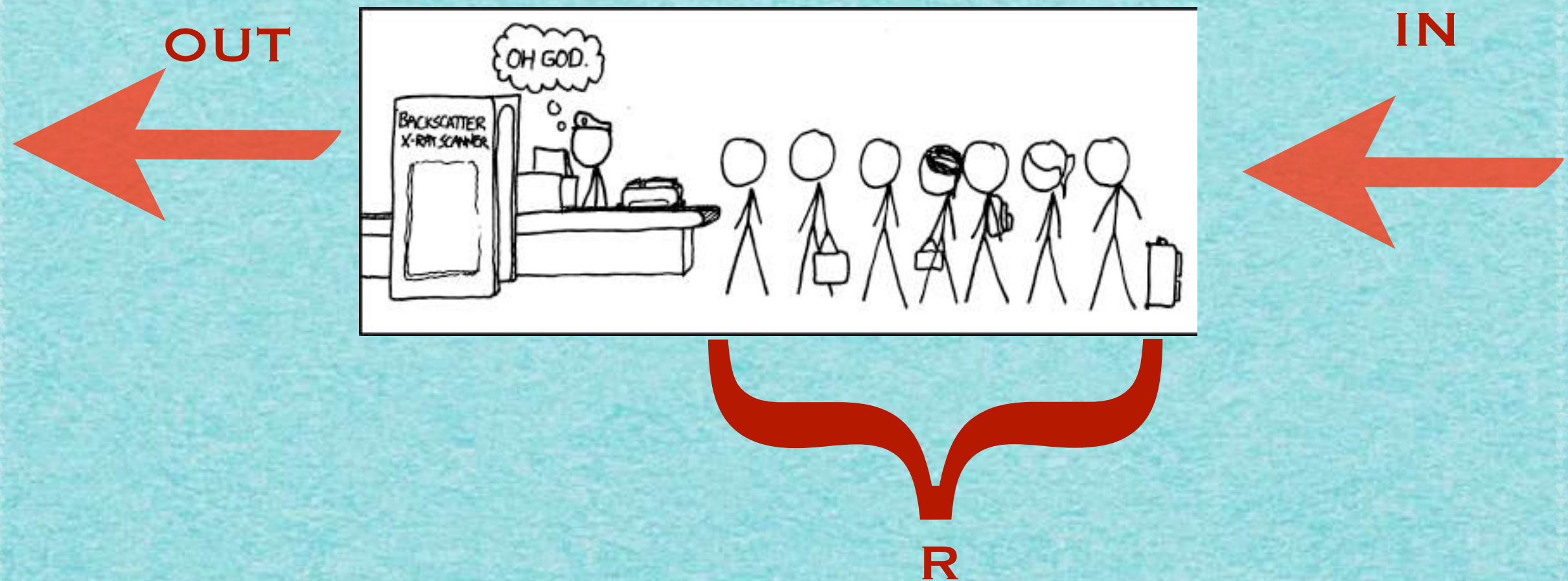
CAN YOU MAKE IT
WITHOUT KILLING
YOURSELF?



Algorithmus

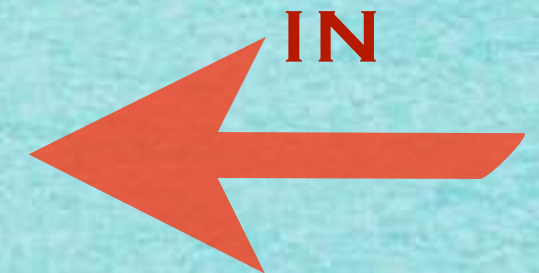
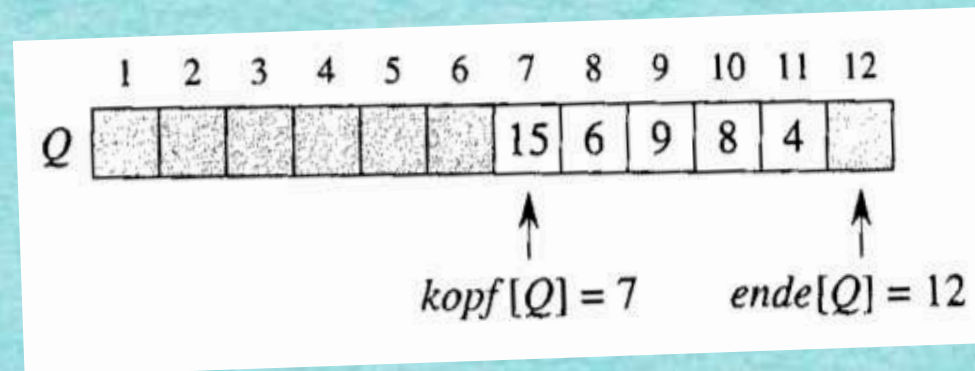
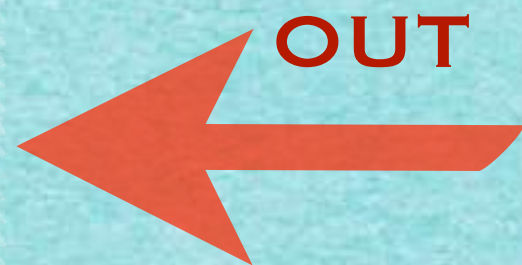
DATENSTRUKTUR

DATENSTRUKTUR I

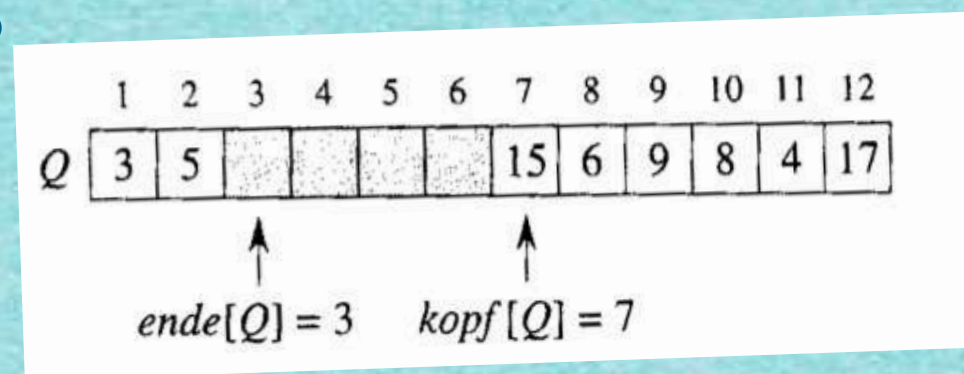


WARTESCHLANGE: FIRST IN - FIRST OUT

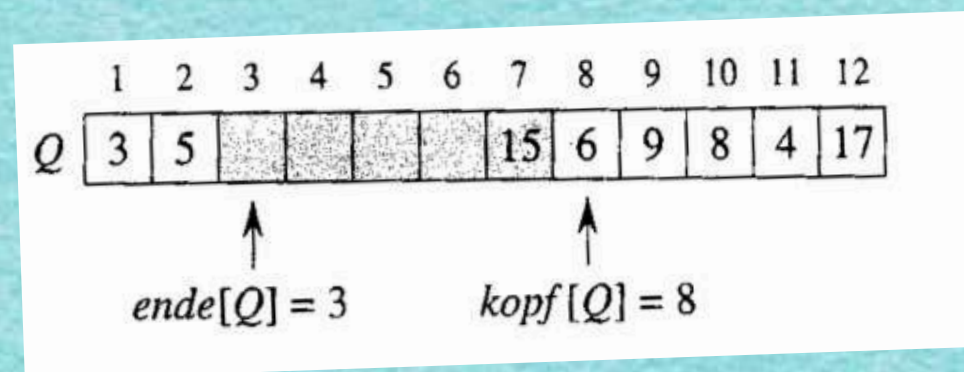
WARTESCHLANGE AUF ARRAY UMGESETZT



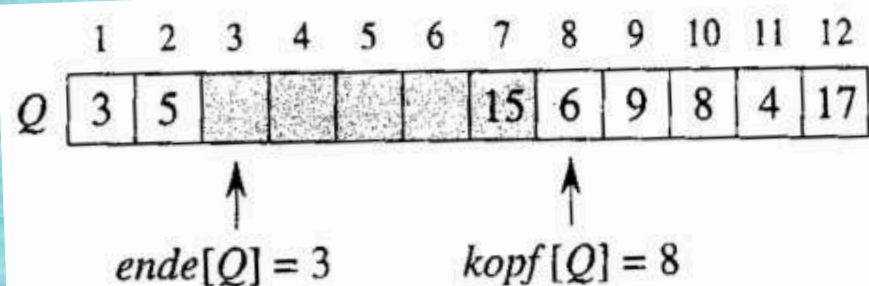
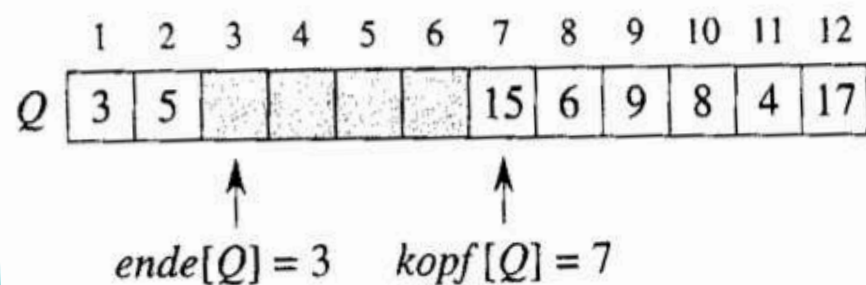
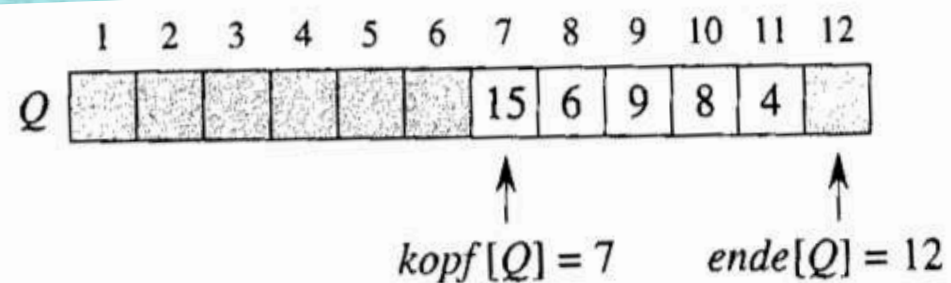
ENQUEUE: 17, 3, 5



DEQUEUE:



WARTESCHLANGE AUF ARRAY UMGESETZT



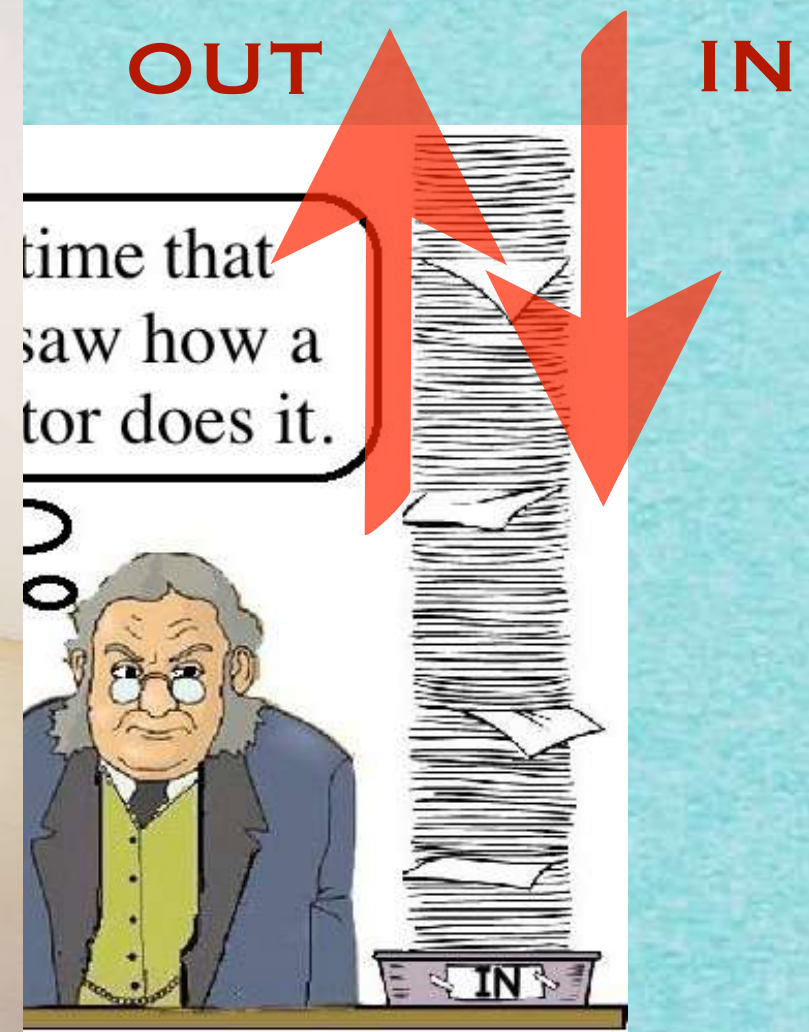
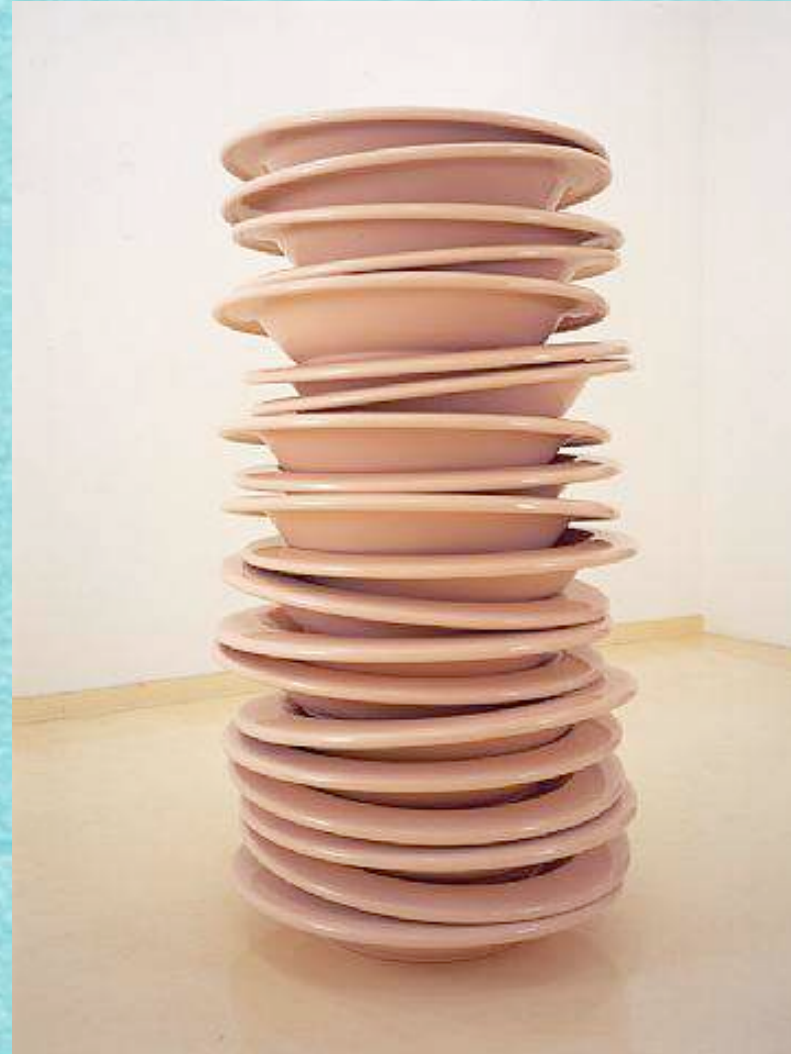
```

4   else  $\text{kopf}[Q] \leftarrow \text{kopf}[Q] + 1$ 
5   return  $x$ 

```


DATENSTRUKTUR II

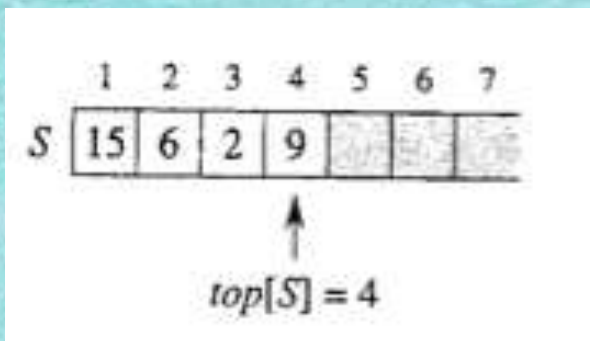
R



STAPEL: LAST IN - FIRST OUT

STACK

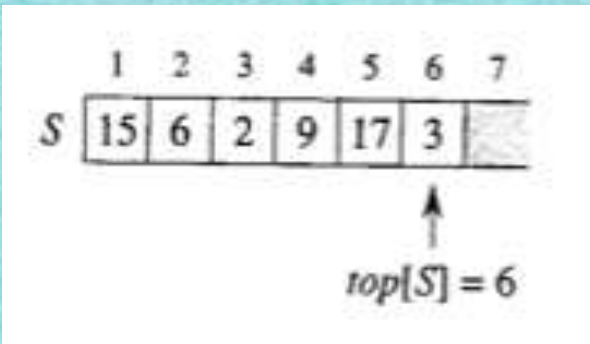
AUF ARRAY UMGESETZT



```

STACK-EMPTY(S)
1  if  $top[S] = 0$ 
2      then return WAHR
3      else return FALSCH
    
```

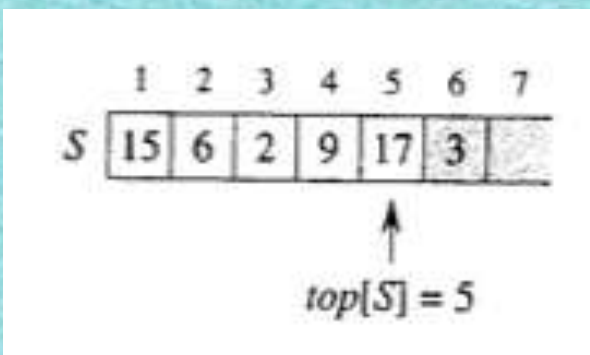
PUSH: 17, 3



```

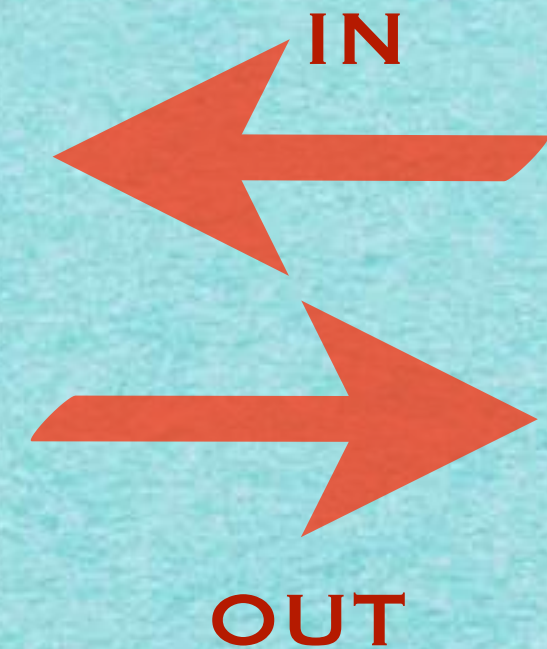
PUSH(S, x)
1   $top[S] \leftarrow top[S] + 1$ 
2   $S[top[S]] \leftarrow x$ 
    
```

POP



```

POP(S)
1  if STACK-EMPTY(S)
2      then error "Unterlauf"
3      else  $top[S] \leftarrow top[S] - 1$ 
4              return  $S[top[S] + 1]$ 
    
```



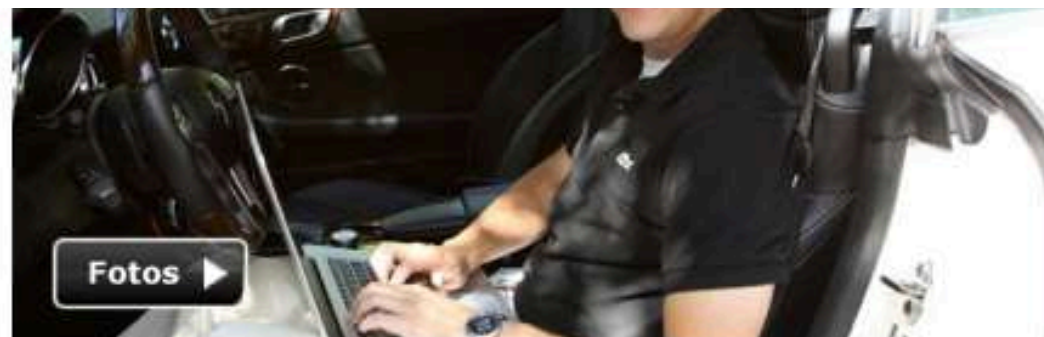
KARRIERECHANCEN!

KARRIERE SPIEGEL

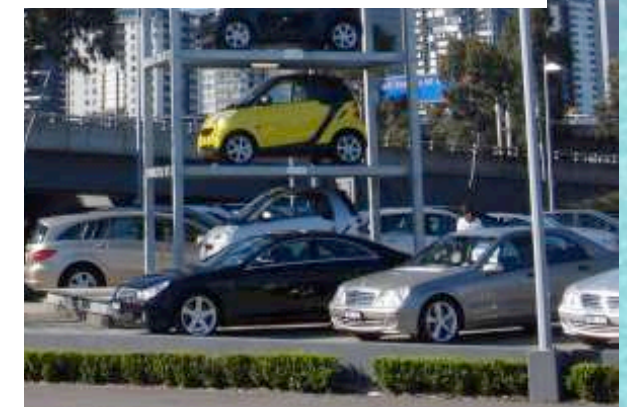
● Home ● Berufsstart ● Berufsleben ● Ausland ● Stellensuche ● Jobs

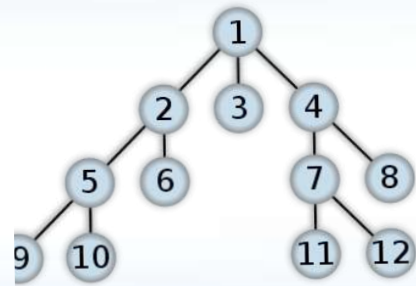
Am liebsten reine Informatiker

Insgesamt 30 Mitarbeiter werkeln bei Daimler an der Systemarchitektur des Infotainment im Auto, die meisten sind Informatiker oder Ingenieure der Elektrotechnik. Vor sechs Jahren waren es noch halb so viele. "Bei Berufseinsteigern sind uns reine Informatiker am liebsten", sagt Matthias Stümpfle, der Leiter der Abteilung: "Sie sind in der Informatik gründlich ausgebildet, das Branchen-Know-how bringen wir ihnen bei."



Daimler



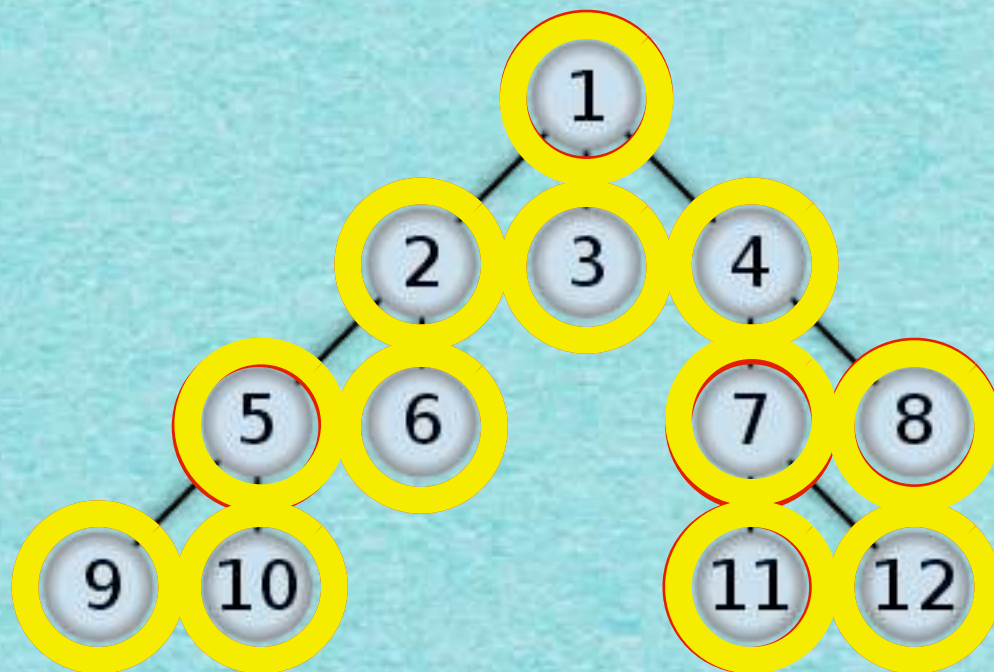


Kapitel 3.5: Tiefensuche und Breitensuche

*Algorithmen und Datenstrukturen
WS 2021/22*

Prof. Dr. Sándor Fekete

Graphenscan mit WARTESCHLANGE

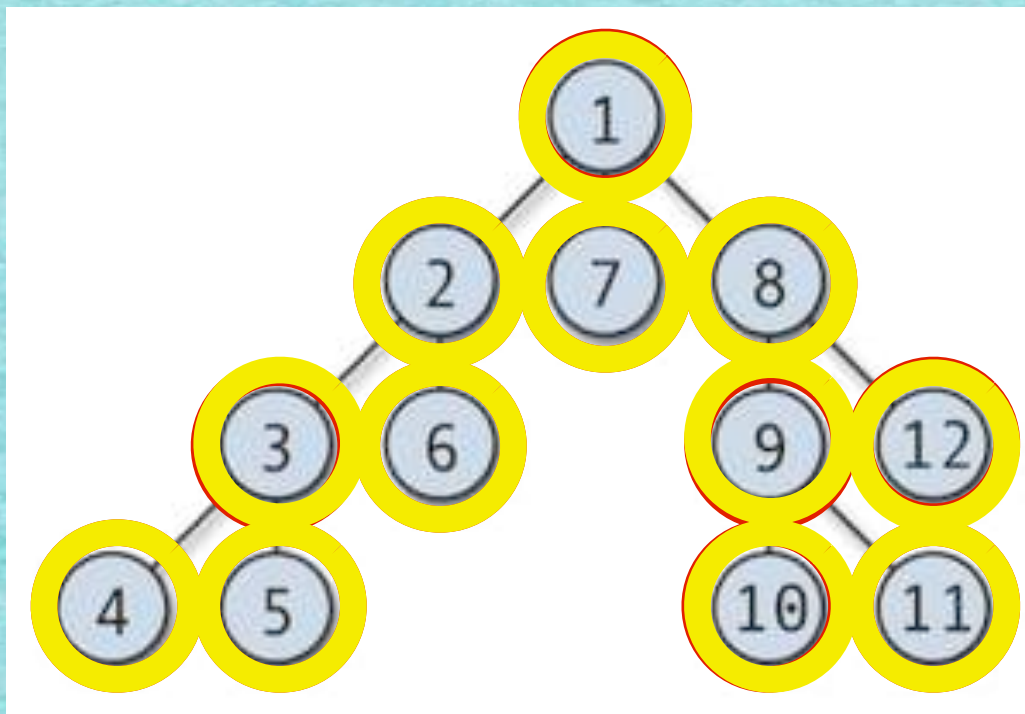


R: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

STOP!

BREITENSUCHE - "BREADTH-FIRST SEARCH" (BFS)

Graphenscan mit STAPEL

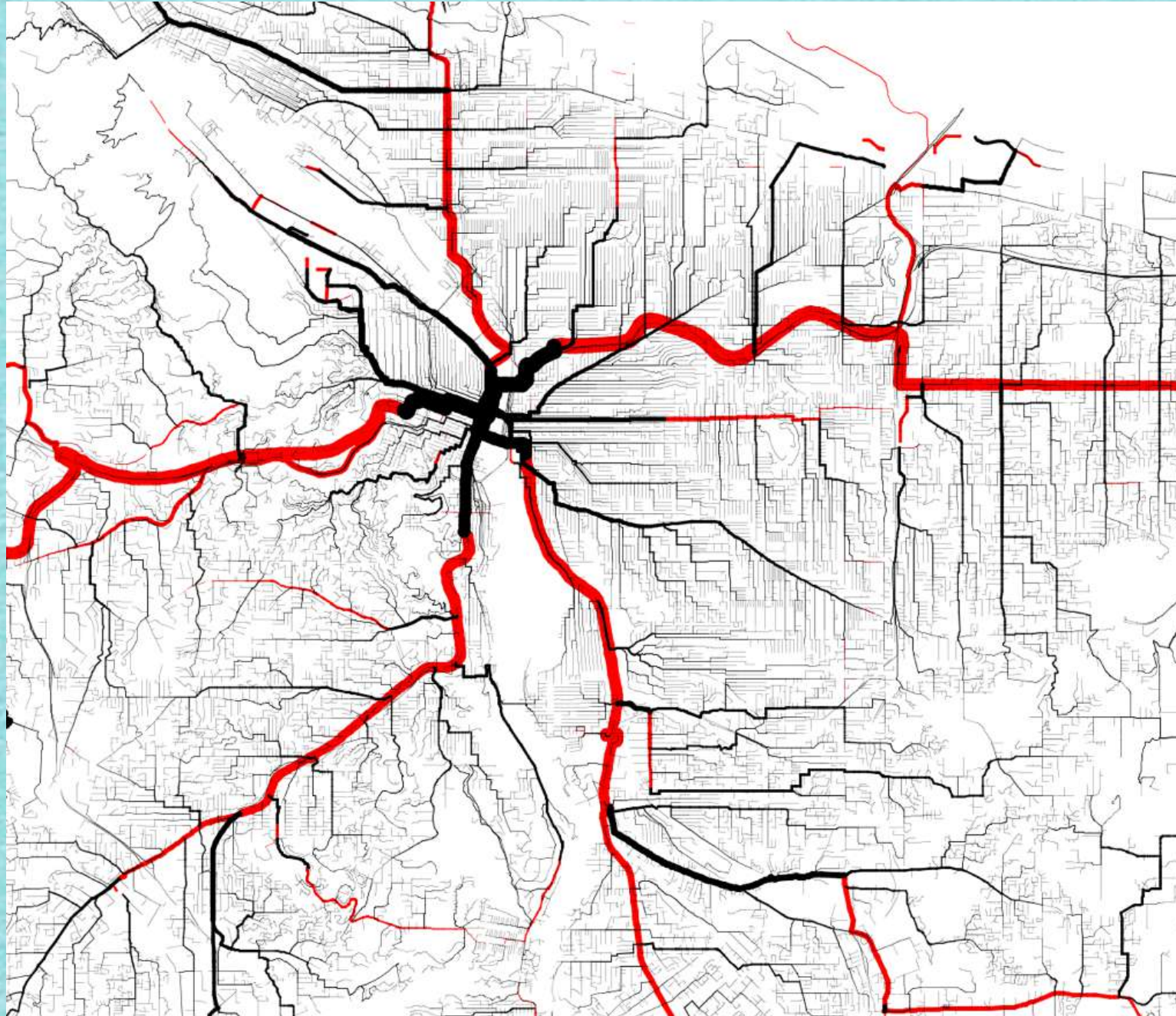


R: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

STOP!

TIEFENSUCHE - “DEPTH-FIRST SEARCH” (DFS)

Breitensuche liefert kürzeste Wege von einer Quelle aus



Viele "Sammler"

Mehr demnächst!

s.fekete@tu-bs.de