

Kapitel 3.5: Tiefensuche und Breitensuche

*Algorithmen und Datenstrukturen
WS 2023/24*

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

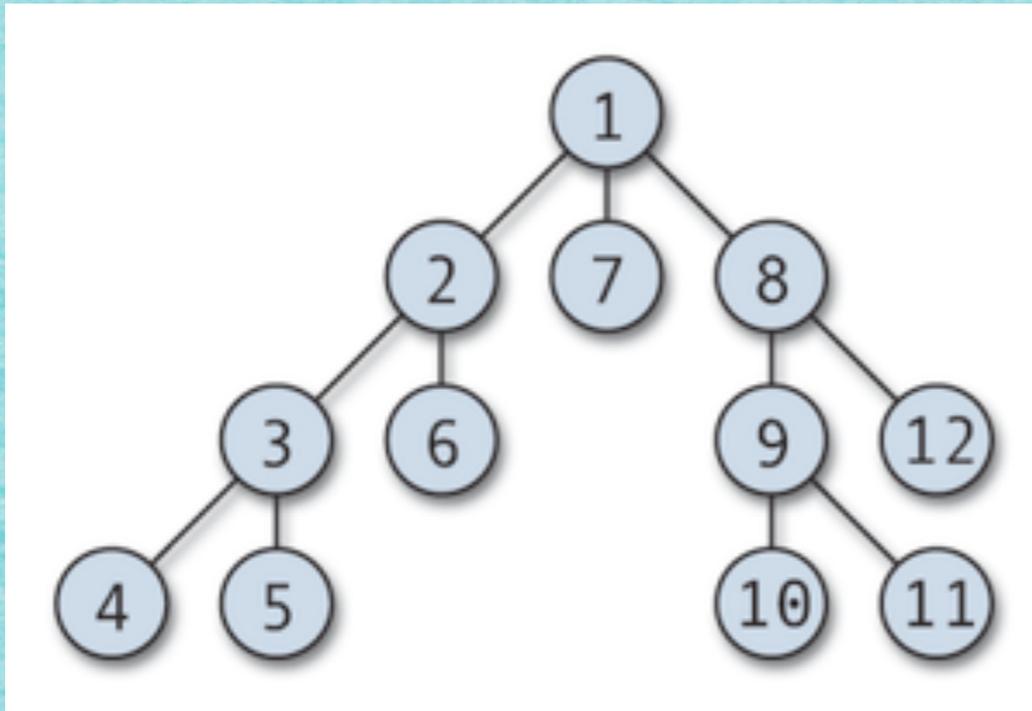
Algorithmus 3.7

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

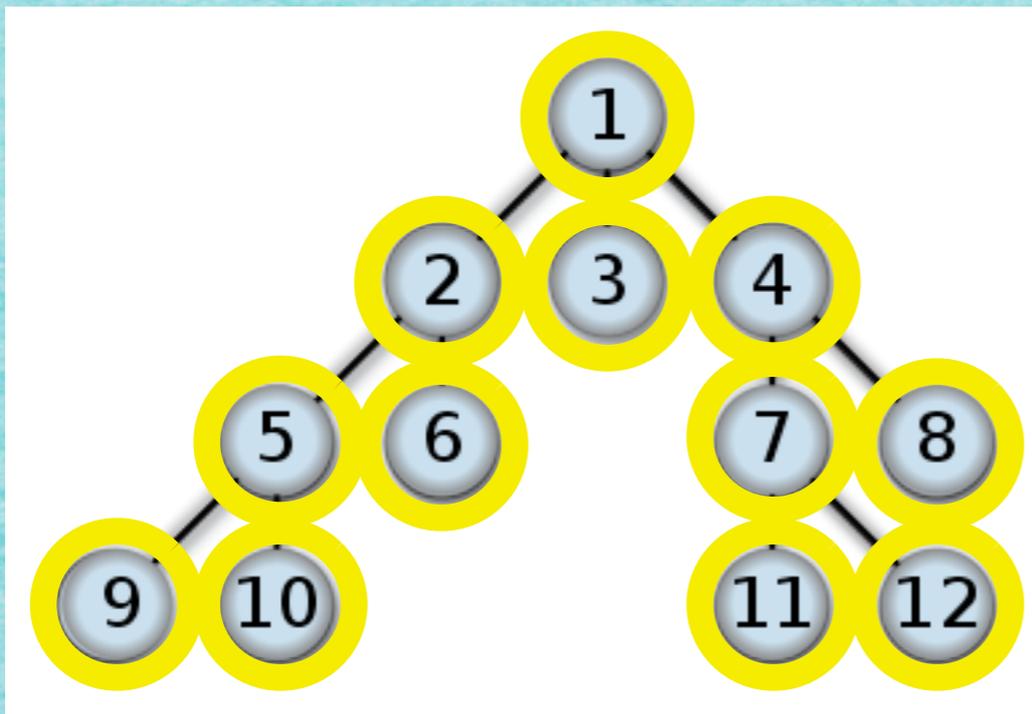
OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,
Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R := \{s\}$, $Y := \{s\}$, $T := \emptyset$
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

Graphenscan mit WARTESCHLANGE



Graphenscan mit WARTESCHLANGE

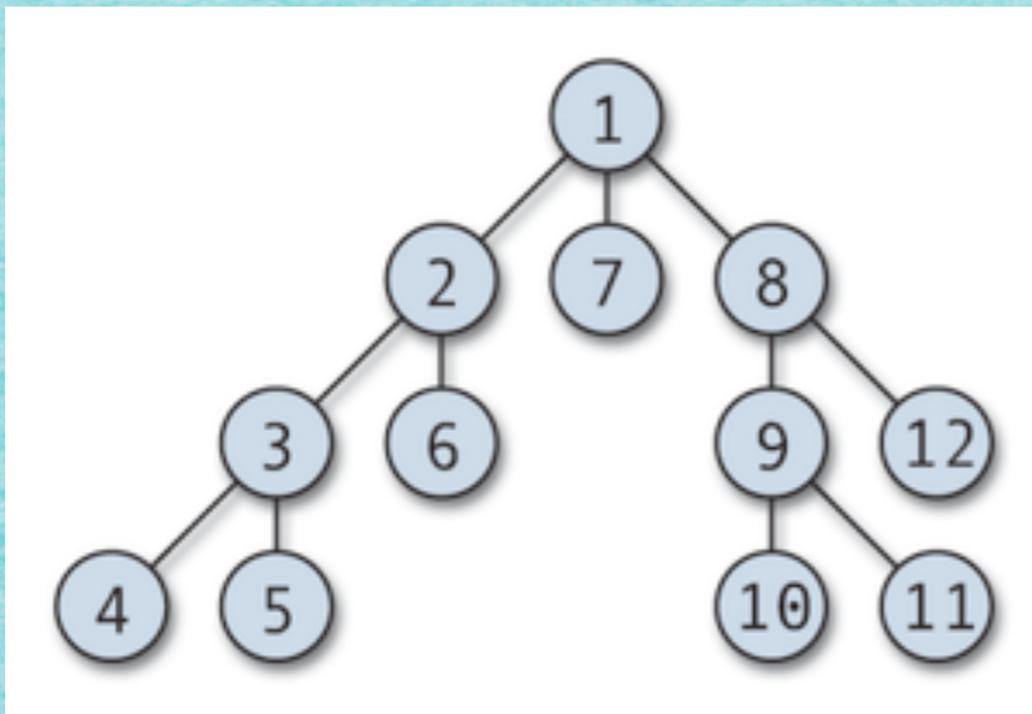


R:

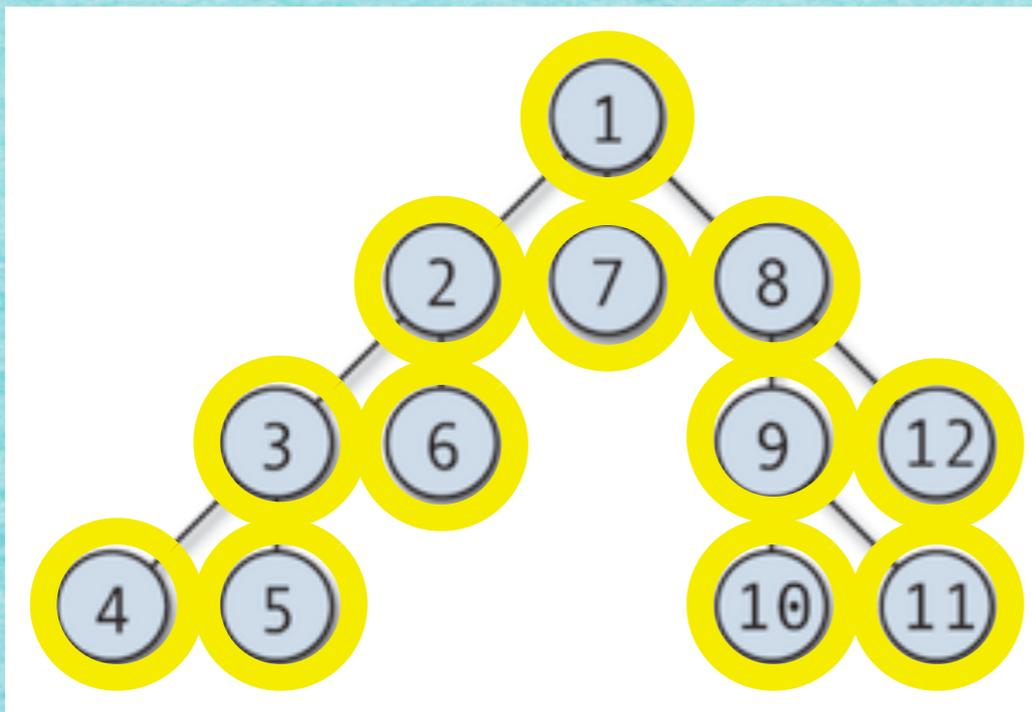
STOP!

BREITENSUCHE - "BREADTH-FIRST SEARCH" (BFS)

Graphenscan mit STAPEL



Graphenscan mit STAPEL



R:

STOP!

TIEFENSUCHE - "DEPTH-FIRST SEARCH" (DFS)

Auf die Schnelle mit der Welle

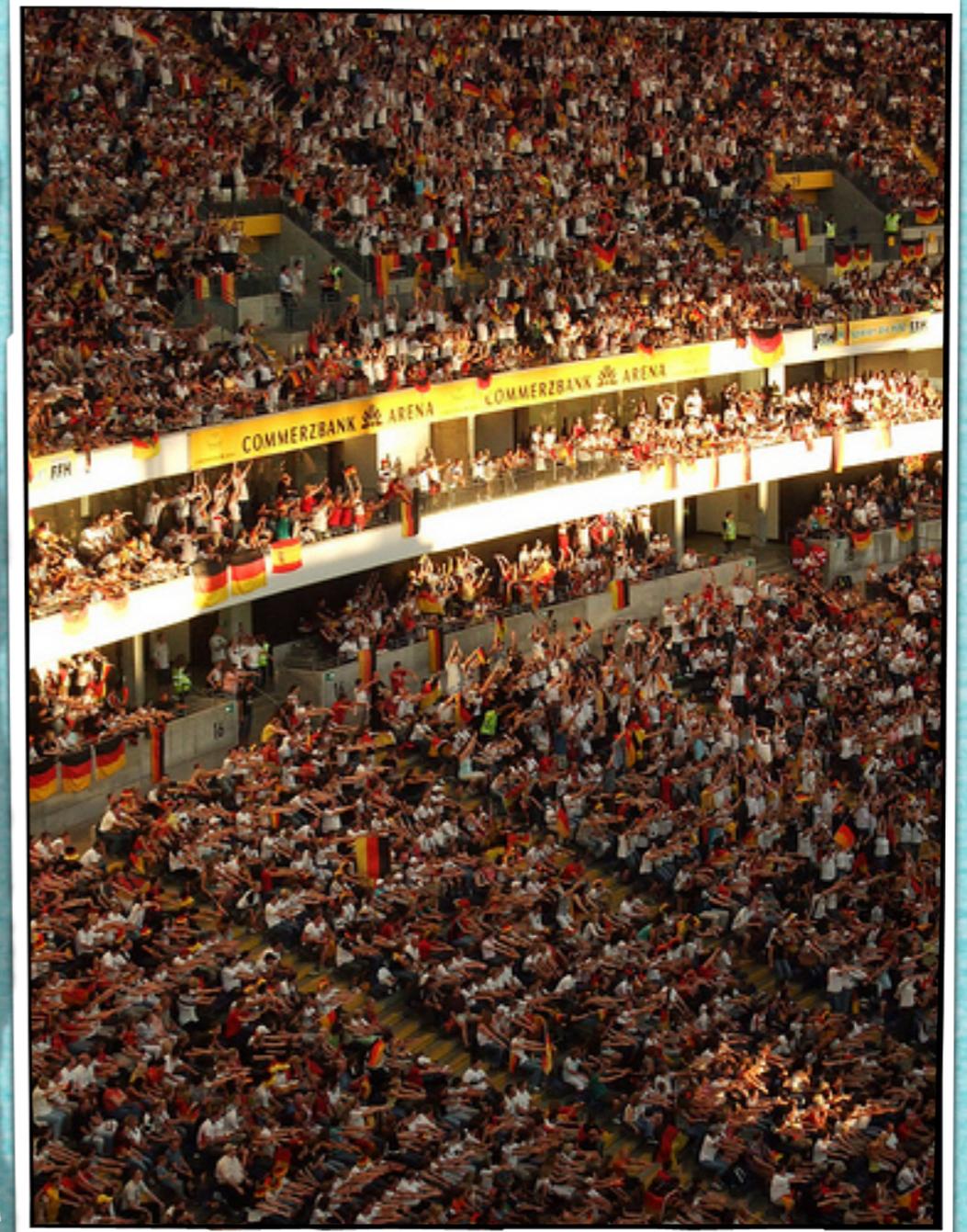
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

- Solange du noch nicht aufgestanden warst:
 - ▶ Wenn ein oder mehrere direkte Nachbarn aufstehen:
 1. Einen dieser Nachbarn merken
 2. In der nächsten Runde:
 - 2.1. aufstehen
 - 2.2. Zahl merken
 3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

- Auf gemerkten Nachbarn zeigen



Auf die Schnelle mit der Welle

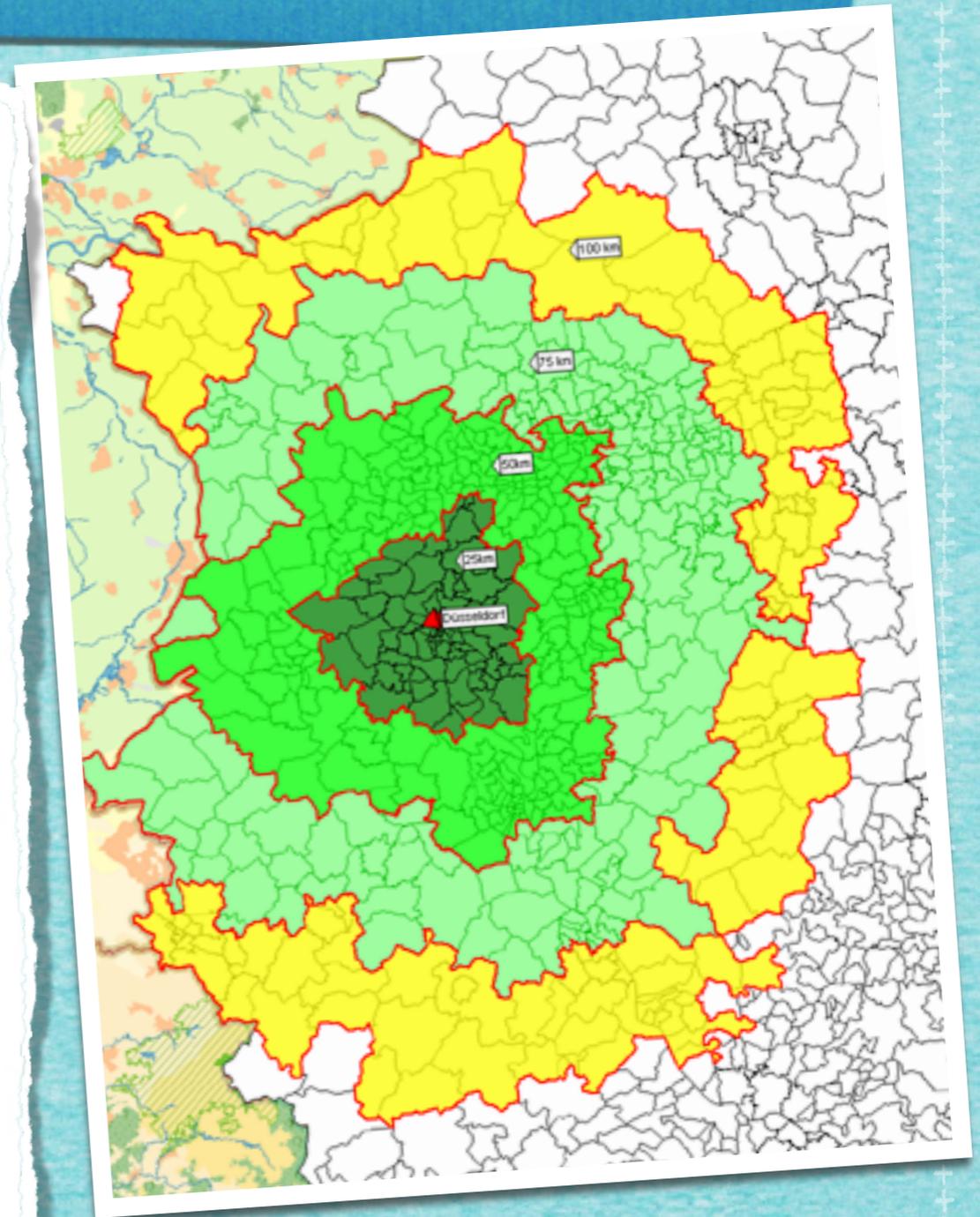
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

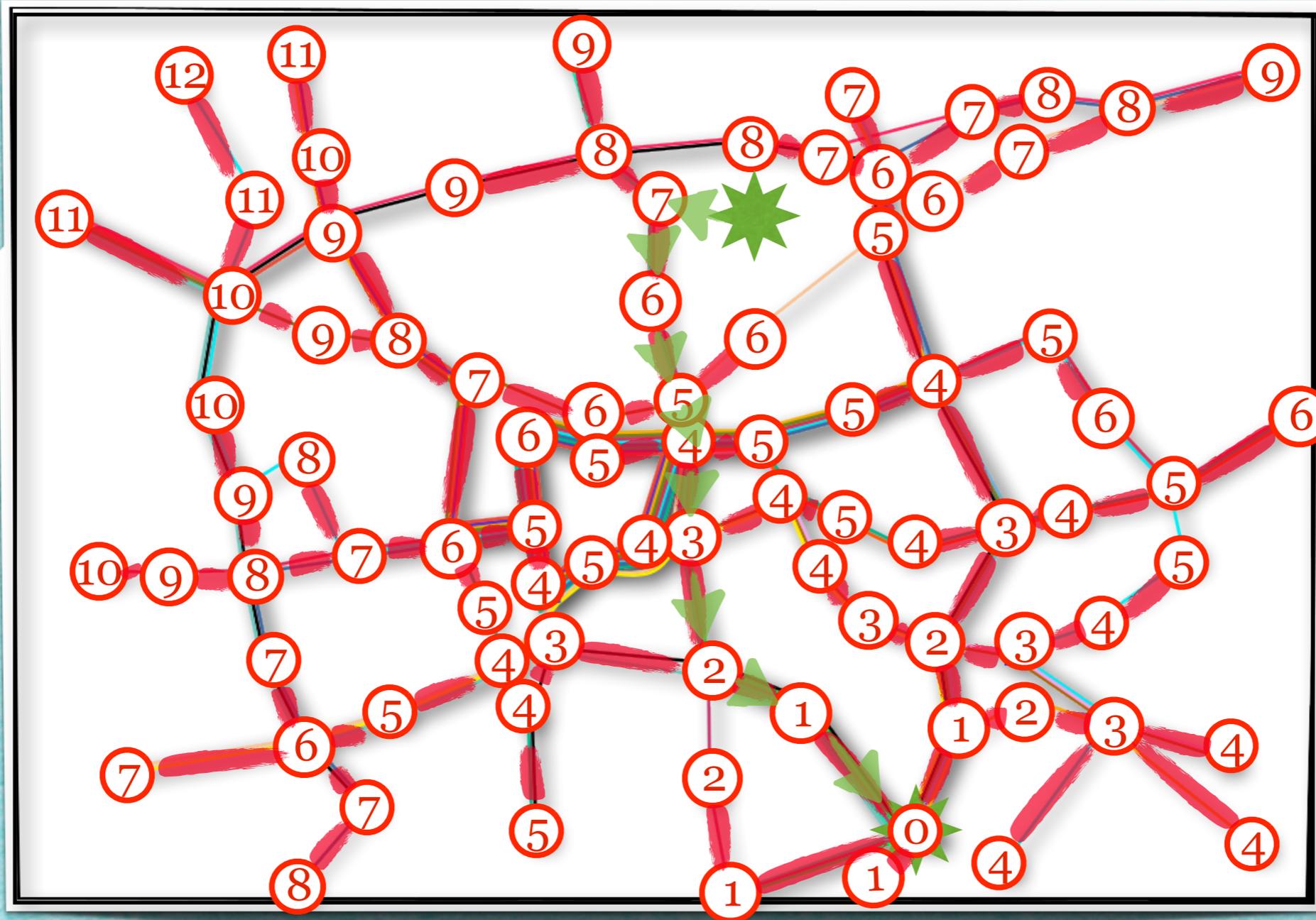
- Solange du noch nicht aufgestanden warst:
 - ▶ Wenn ein oder mehrere direkte Nachbarn aufstehen:
 1. Einen dieser Nachbarn merken
 2. In der nächsten Runde:
 - 2.1. aufstehen
 - 2.2. Zahl merken
 3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

- Auf gemerkten Nachbarn zeigen

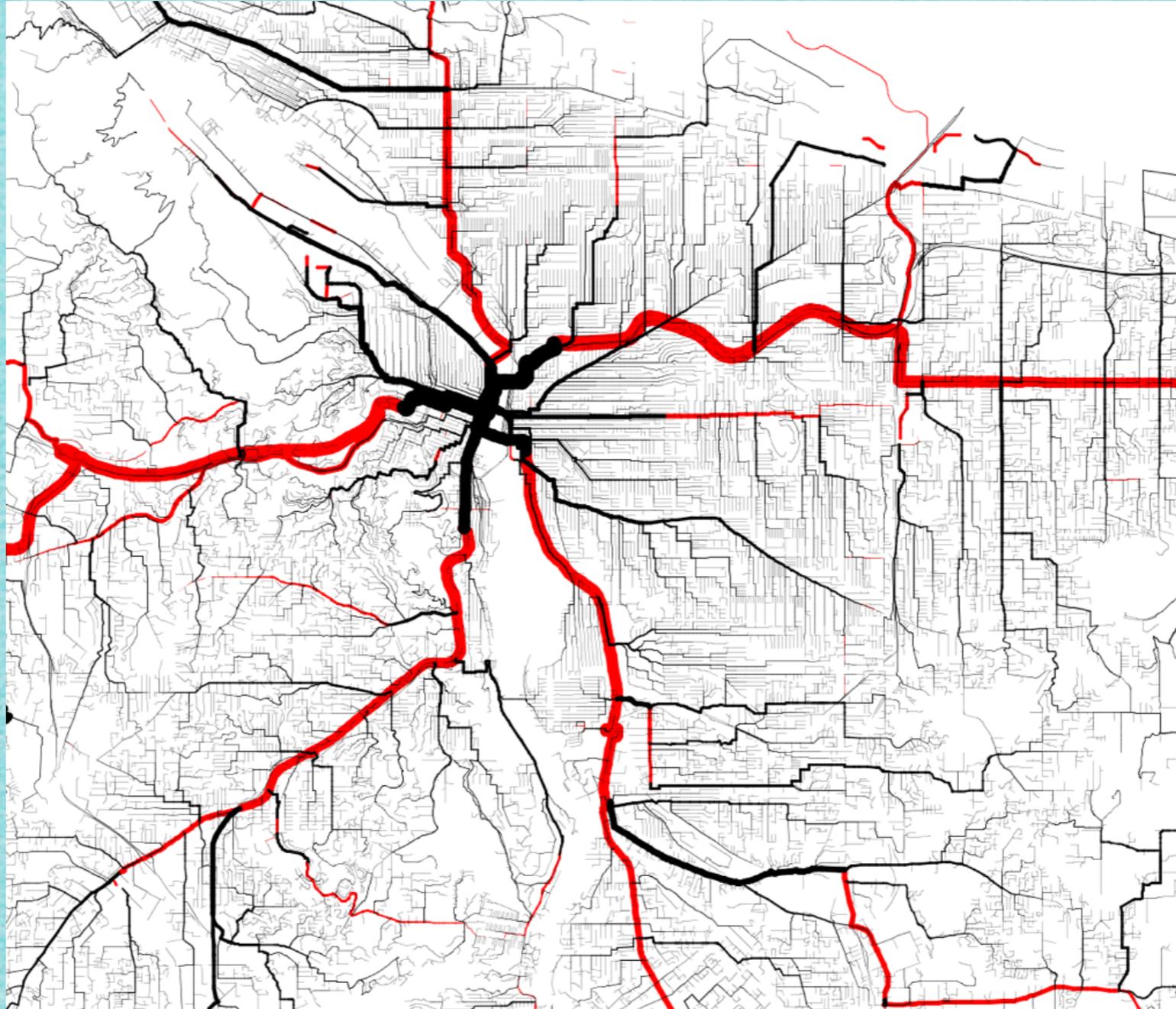


Wellenreiten in Graphen



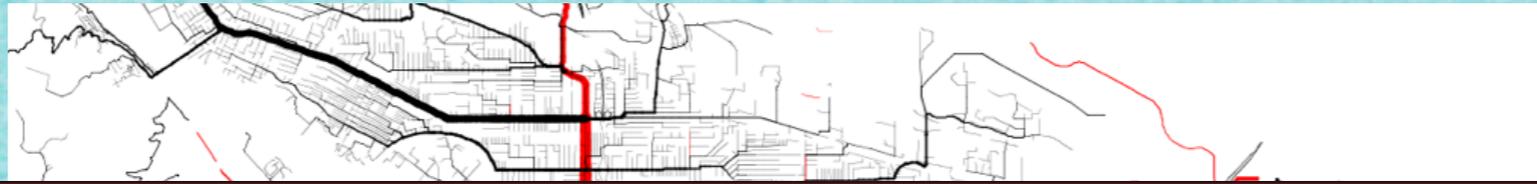
Breitensuche

Breitensuche liefert kürzeste Wege von einer Quelle aus



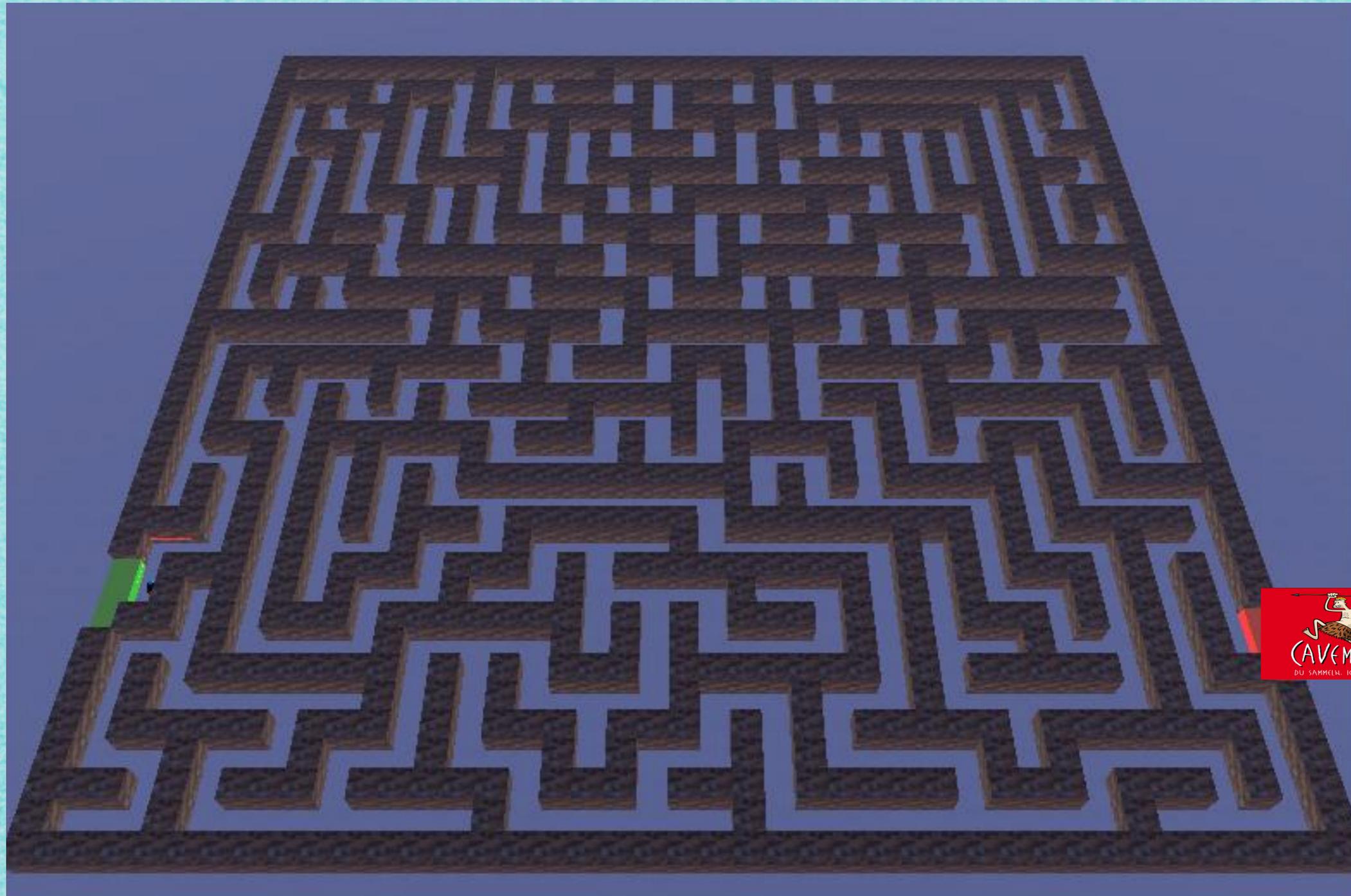
Viele "Sammler"

Breitensuche liefert kürzeste Wege von einer Quelle aus



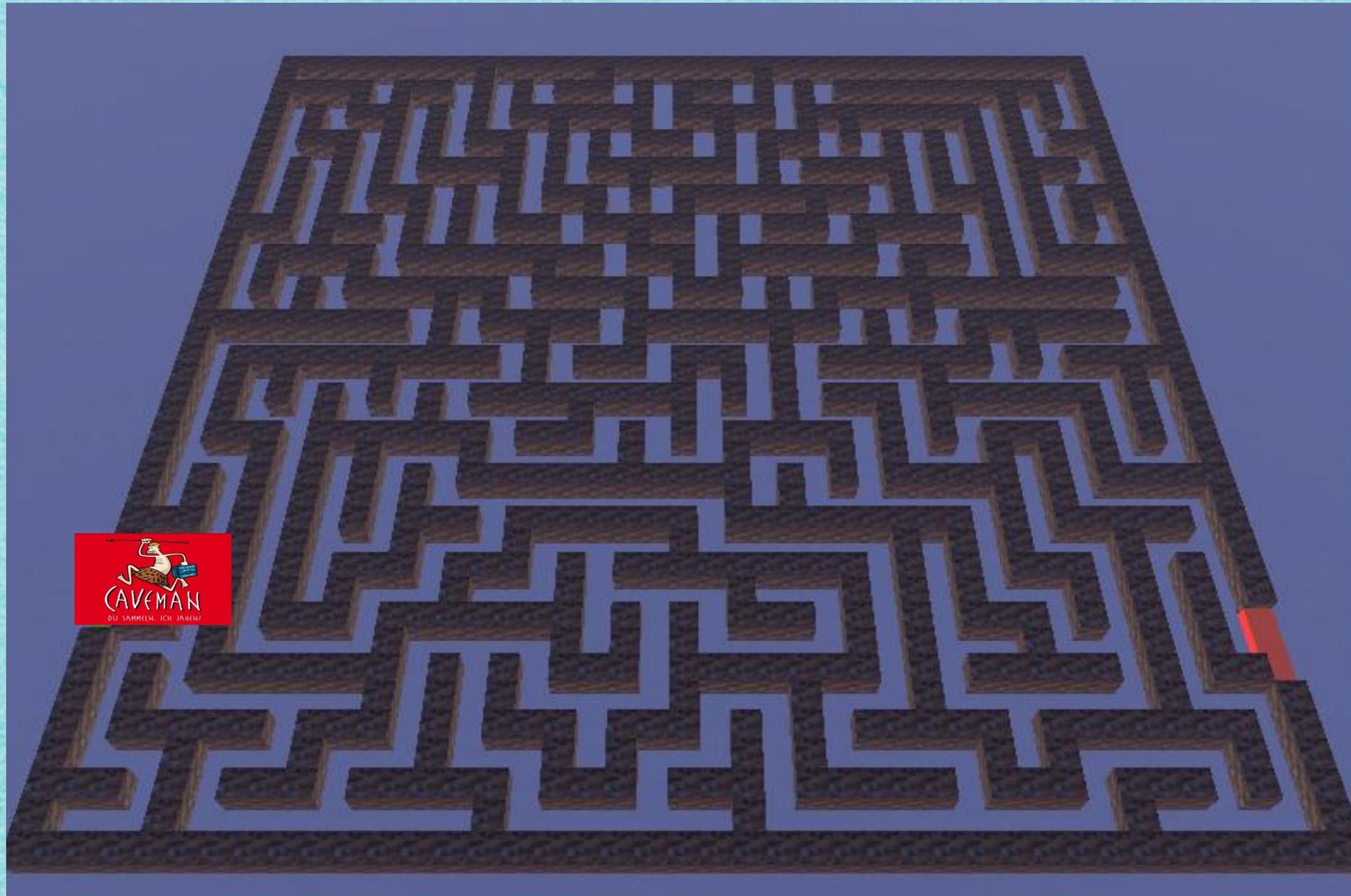
Viele "Sammler"

Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"

Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"



Zum dritten Mal ein Paar

Bettina und Christian Wulff zeigen sich erstmals wieder gemeinsam

Der ehemalige Bundespräsident Christian Wulff und seine Frau Bettina wagen zum dritten Mal einen Neustart ihrer Beziehung. Nun ist das Paar gemeinsam bei einem Presse-Event aufgetreten.

22.06.2021, 19.17 Uhr





"I told you we should have asked for directions."

WENN NAVIGATIONSSYSTEME WIRKLICH WEIBLICH WÄREN...

NA GUT, DANN KANN ICH HALT KEINE KARTEN LESEN!

ABER WER WOLLTE DENN NICHT ANHALTEN UND NACH DEM WEG FRAGEN?!



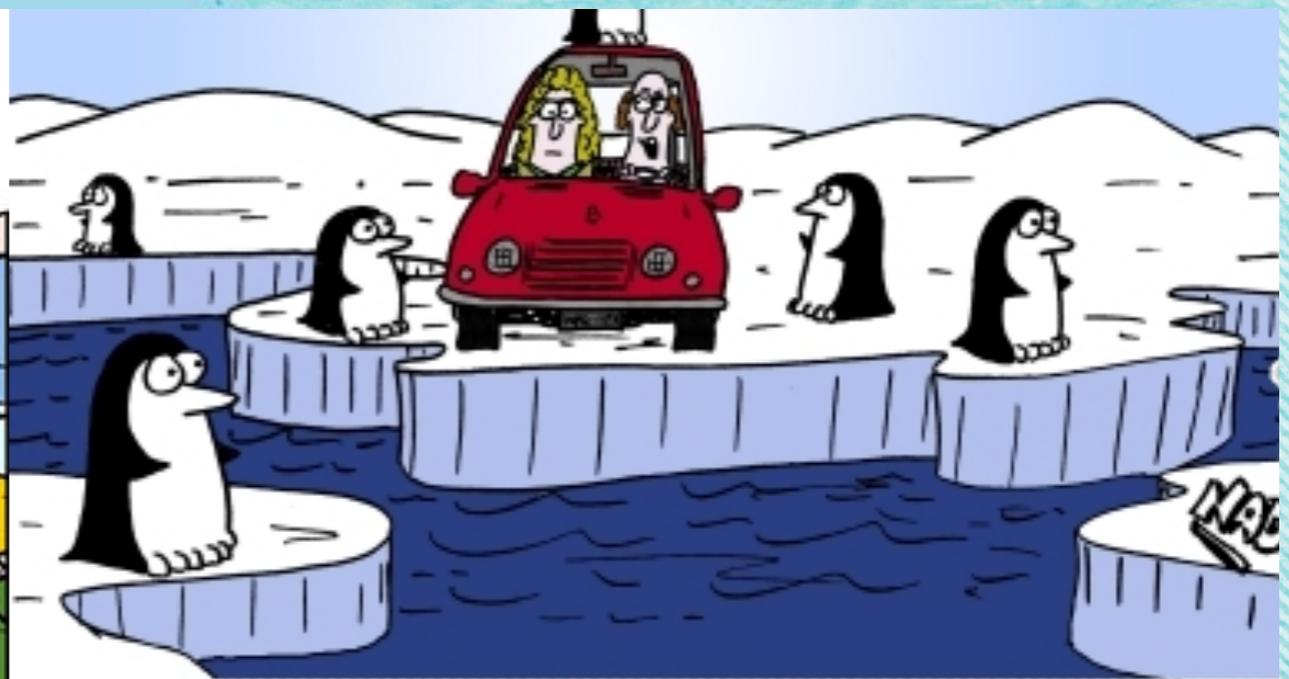
MYCARTOONS.DE

WHY DID MOSES WANDER IN THE DESERT FOR FORTY YEARS?

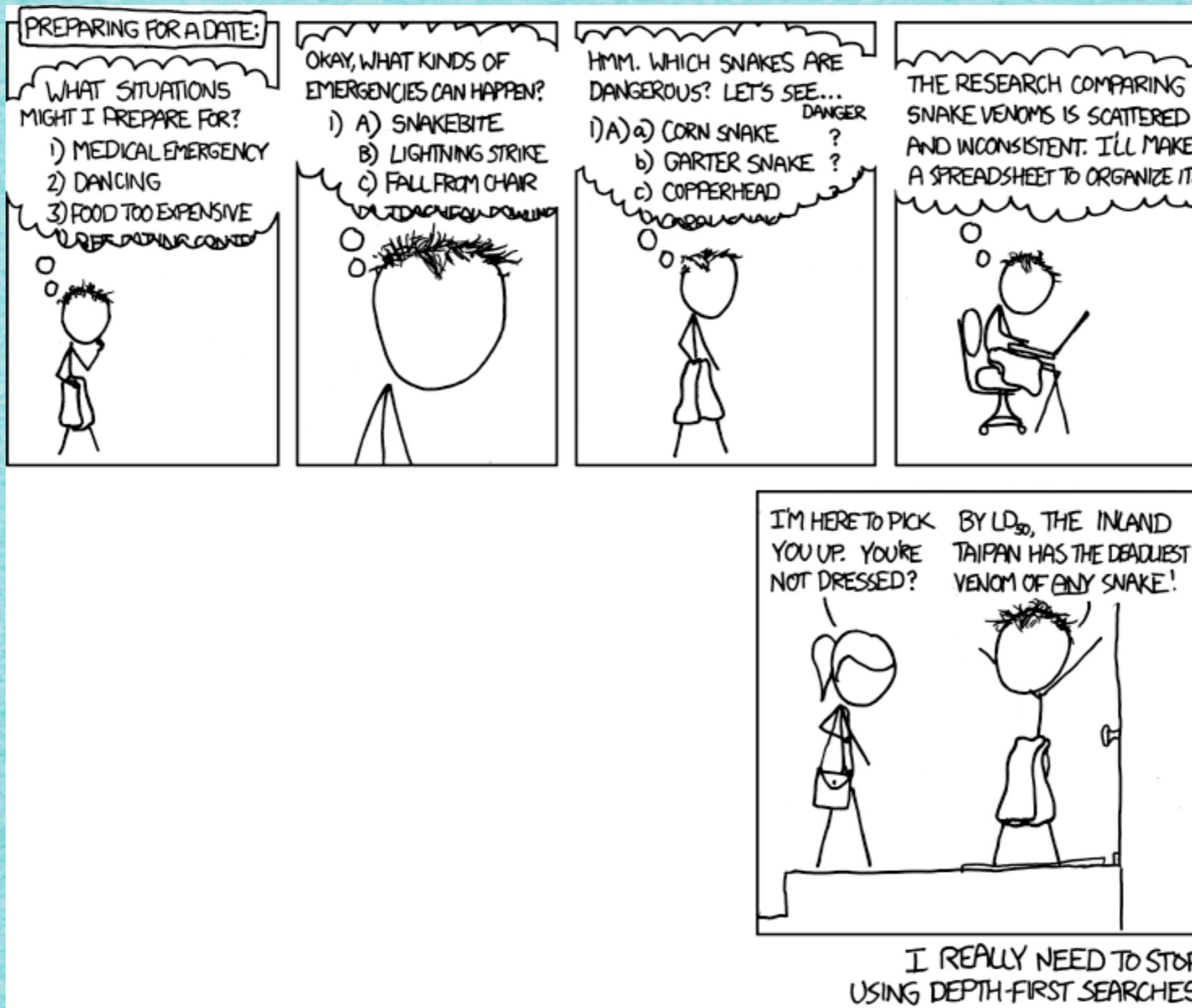
BECAUSE EVEN BACK THEN A MAN WOULDN'T STOP AND ASK FOR DIRECTIONS!



NOW WILL YOU ASK FOR DIRECTIONS?

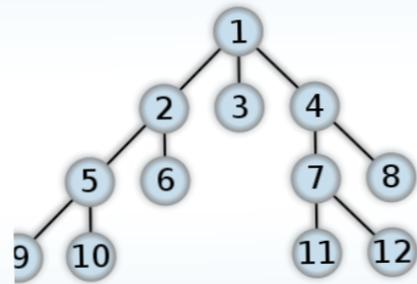


"Alright then, I'll admit it... we're lost!"



Quiz!

s.fekete@tu-bs.de



Kapitel 3.6: Datenstrukturen für Graphen

*Algorithmen und Datenstrukturen
WS 2023/24*

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 $R := \{s\}$, $Y := \{s\}$, $T := \emptyset$
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

Weiter an der Tafel!

s.fekete@tu-bs.de