

Nachbarn
von A

Entf.	Labels
0	A
1	B, D, F
2	C, G
3	B, D
4	C, G
5	B, D
6	C, G
7	B, D

Nachbarn in
Entfernung n:

Tabelle bis
Zeile n
entwickeln
und diese
Zeile liefern.

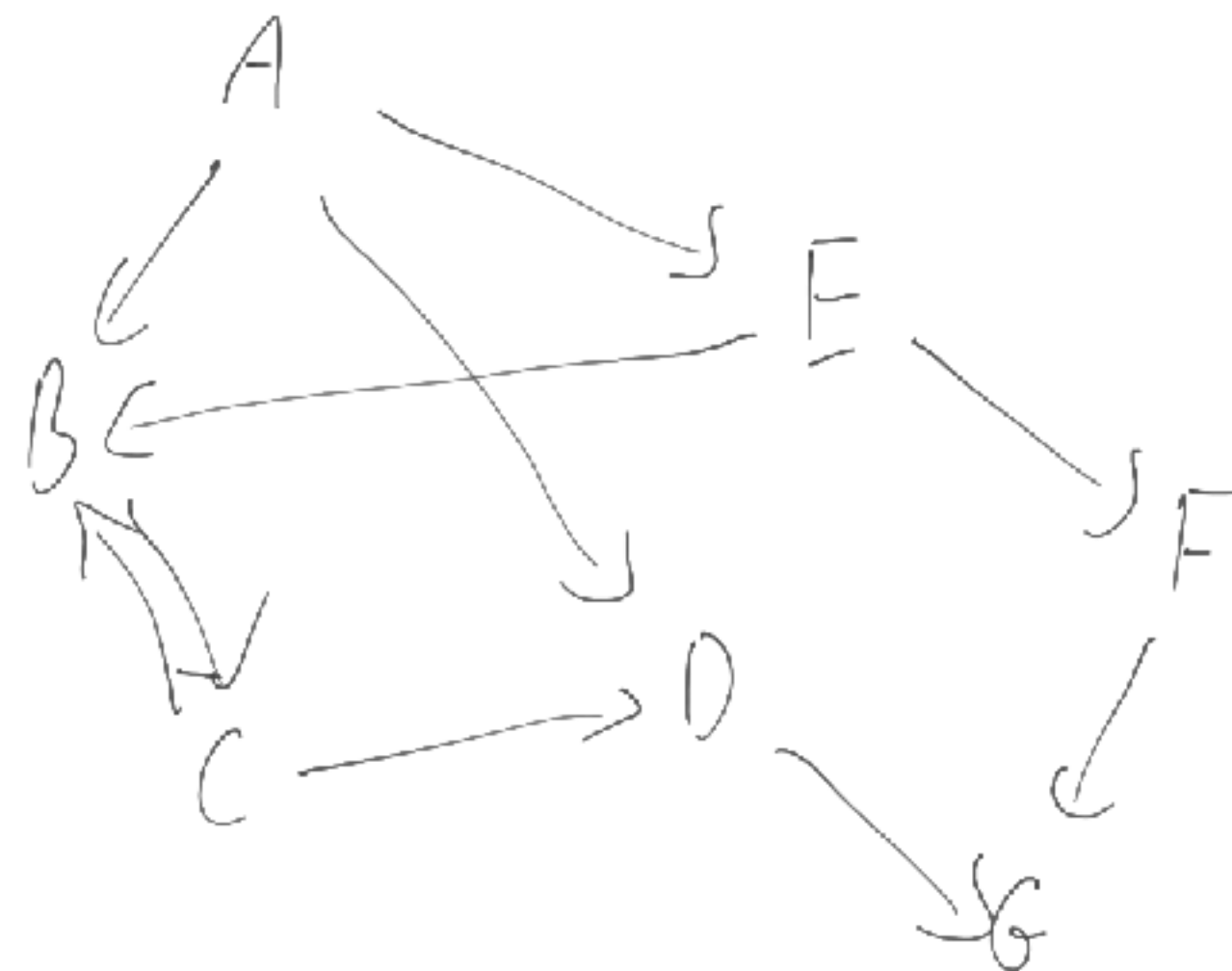
Kürzeste
Entfernung zu
einem Knoten
bestimmen:

Tabelle entwickeln
und mitzählen.
Abbrechen, sobald
der Zielknoten
vorkommt.

Erreichbarkeit
prüfen:

Wie beim
Entfernung messen.
Wenn nach
<Knotenzahl>
Schritten nicht
erreicht, dann False.

Kürzeste Wege finden



Zwei
Listen von
Knoten
verwalten:

Schon
besuchte
Knoten

Aktuelle
Knoten
("Frontier")

1	2	3	4
		B	C
	A	D	G
		E	F
A	B	C	(leer) → fertig
D		G	
E		F	

Algorithmus
von Dijkstra.

A*-Algorithmus

Ergebnis:

A → B → C
A → D → G
A → E → F

Breitensuche

Wir
durchsuchen
den gesamten
Graphen
"gleichzeitig".

Erläuterung
der
O-Notation

**Binäre
Suche
ist in**

$$O(\log n)$$

**MergeSort
ist in**

$$O(n \cdot \log n)$$

**BubbleSort ist
in**

$$O(n^2)$$

Bedeutung:

**ist nicht
schlimmer als**

**wird nach
oben
beschränkt
von**

$$\exists C > 0 \exists x_0 > 0 \forall x > x_0 : |f(x)| \leq C \cdot |g(x)|$$

$f \in O(g)$, wenn

... es eine
Zahl C
gibt ...

... und es
einen
Startwert
 x_0 gibt ...

... so dass
für alle x
rechts von
 x_0 gilt:

$$f(x) < C \cdot g(x)$$

$f \in O(n^4)$

... es eine
Zahl C
gibt ...

... und es
einen
Startwert
 n_0 gibt ...

... so dass
für alle n
rechts von
 n_0 gilt:

$$f(n) < n^2$$