Karlsruher Institut für Technologie

Vertex Ordering Problems

Lecture · May 14, 2018
Markus Schneckenburger

INSTITUT FÜR THEORETISCHE INFORMATIK · LEHRSTUHL ALGORITHMIK

Übersicht

Einführung

- Problemstellung
- Notation

1.Algorithmus: Dynamische Programmierung

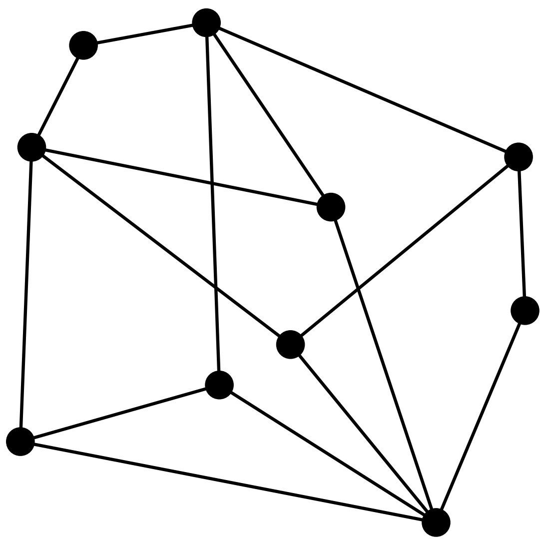
- Beweis
- Pseudocode
- Beispiel

2.Algorithmus: Rekursion

- Pseudocode
- Beispiel

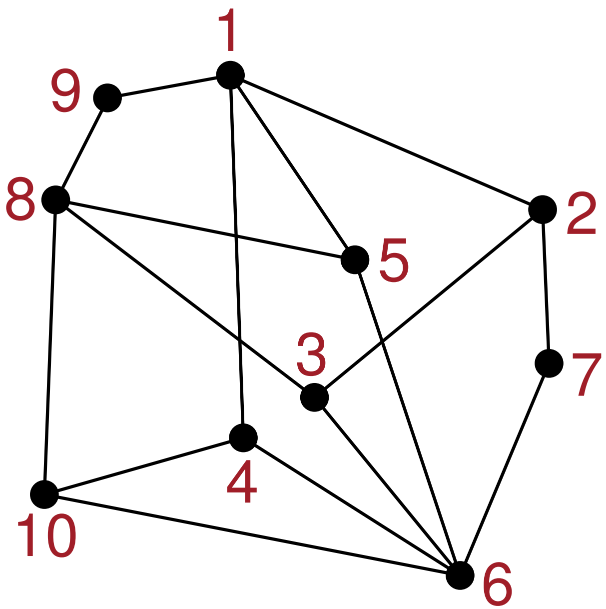
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Problemstellung



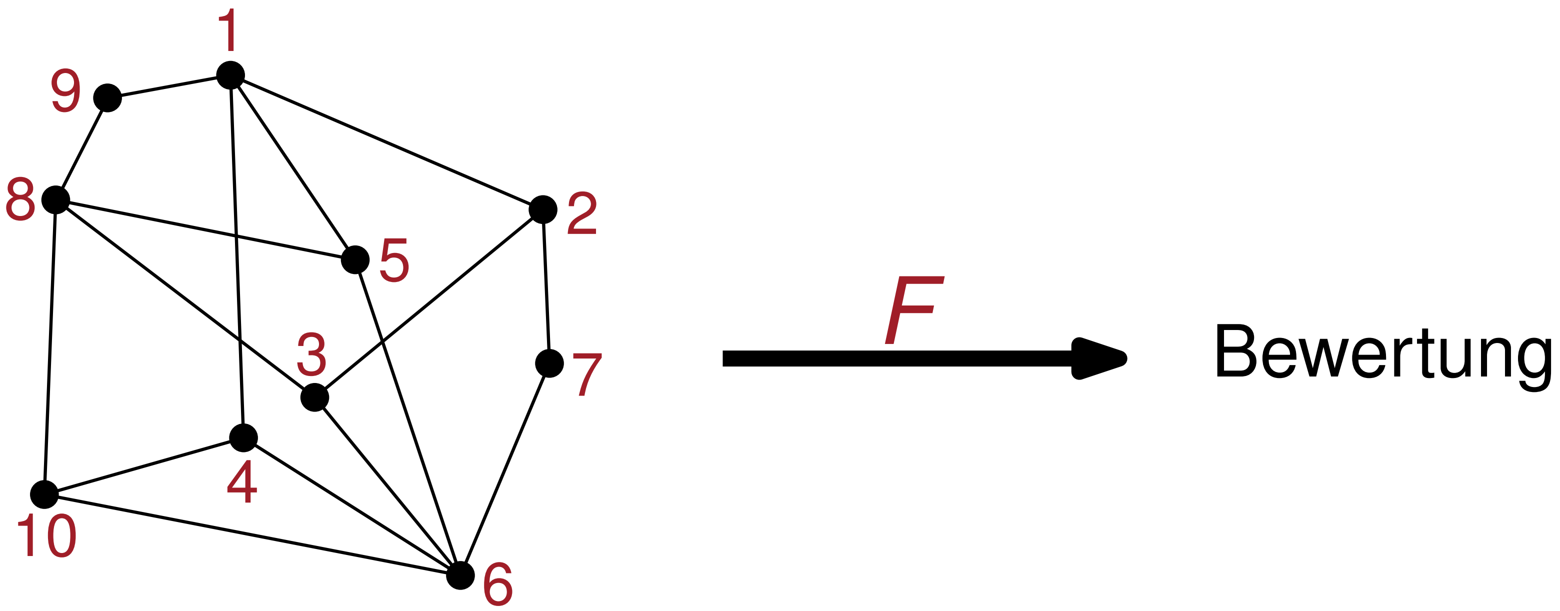
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Problemstellung



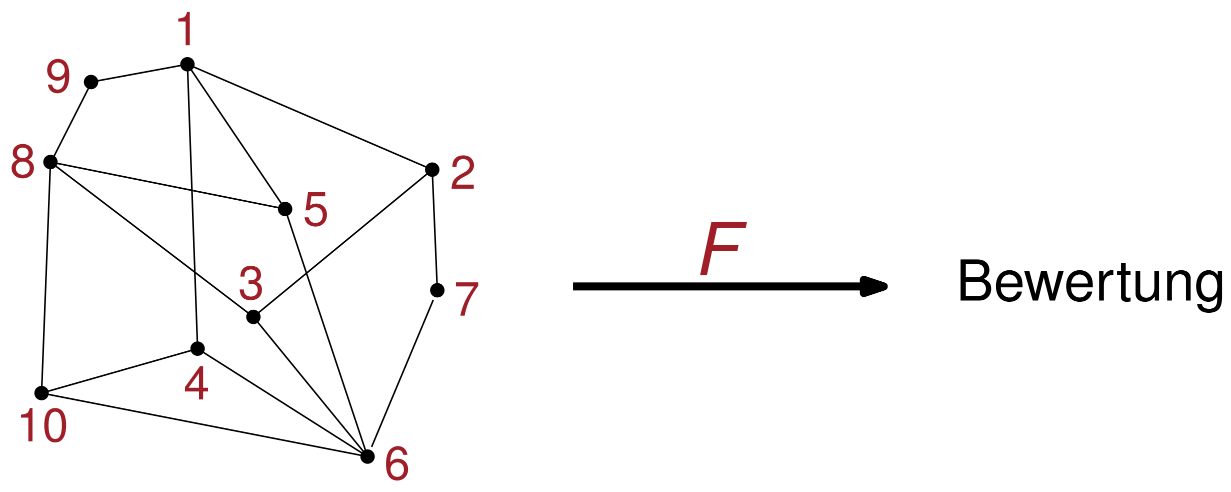
Institut für Theoretische Informatik
Lehrstuhl Algorithmik

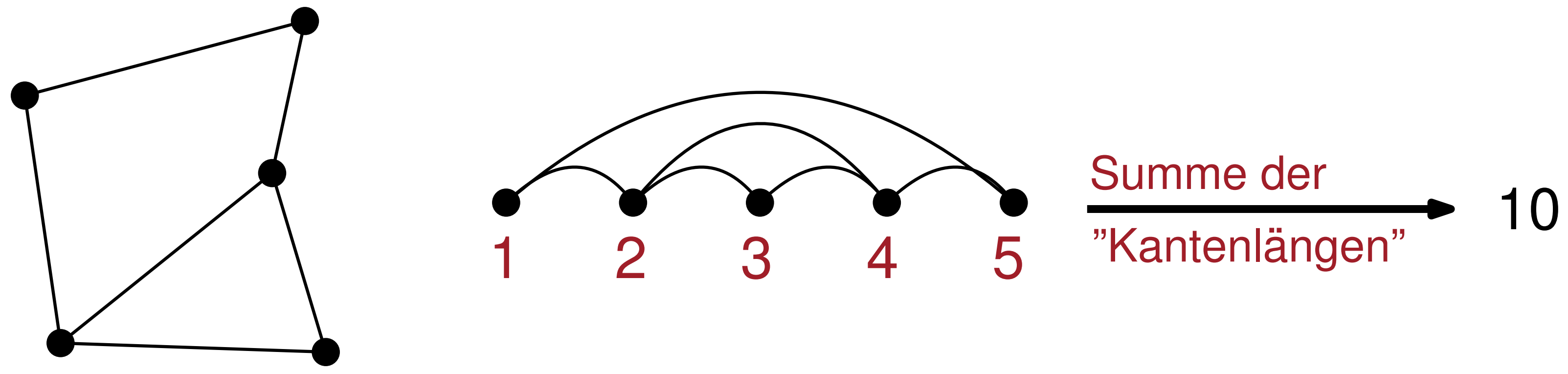
Problemstellung



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Problemstellung





Notation

Definition: Vertex Ordering/Knotenreihenfolge

Ein Vertex Ordering auf einem Graphen $G = (V, E)$ ist eine Bijektion $\pi : V \to \{1, 2, .., |V|\}$

- $\pi_{<v}$ ist die Menge der Knoten, die vor v in der Reihenfolge vorkommen
- $P(V)$ bezeichne die Menge aller Ordering auf dem Graphen $G = (V, E)$
- Für die Laufzeiten: O^* vernachlässigt polynomielle Faktoren

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Problemstellung (Teil 2)

Problemform

Gegeben ein Graph $G = (V, E)$ und eine Funktion $f : (G, L \subseteq V, v \in V) \mapsto r \in \mathbb{R}$, berechne

$$\min_{\pi \in P(V)} \max_{v \in V} f(G, \pi_{<v}, v)$$

oder

$$\min_{\pi \in P(V)} \sum_{v \in V} f(G, \pi_{<v}, v)$$

- Weitere Bedingung: f ist in Polynomialzeit berechenbar
- Kein Zugriff auf die Reihenfolge π in f, aber Wissen über Vorgänger und Nachfolger

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

1. Algorithmus
Dynamische Programmierung

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Karlsruher Institut für Technologie

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$

$$F_G(\emptyset) = -\infty$$

- Wir wollen $F_G(V)$ berechnen

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\varnothing) = -\infty$$

Lemma

Sei $G = (V, E)$ und $S \subseteq V$, $S \neq \varnothing$, dann gilt

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

- Wahl des ”besten” letzten Knoten für die Reihenfolge
- Beweis: Betrachte $\geq$ und $\leq$ separat

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\geq$:

- Wähle $\pi \in P(S)$ so, dass $F_G = \max_{v \in S} f(G, \pi_{<v}, v)$
- Sei w der letzte Knoten in der Reihenfolge, also $\pi(w) = |S|$, dann ist $\pi_{<w} = S \setminus \{w\}$

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\geq$:

- Wähle $\pi \in P(S)$ so, dass $F_G = \max_{v \in S} f(G, \pi_{<v}, v)$
- Sei w der letzte Knoten in der Reihenfolge, also $\pi(w) = |S|$, dann ist $\pi_{<w} = S \setminus \{w\}$

$$F_G(S) = \max_{v \in S} f(G, \pi_{<v}, v)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\geq$:

- Wähle $\pi \in P(S)$ so, dass $F_G = \max_{v \in S} f(G, \pi_{<v}, v)$
- Sei w der letzte Knoten in der Reihenfolge, also $\pi(w) = |S|$, dann ist $\pi_{<w} = S \setminus \{w\}$

$$\begin{aligned} F_G(S) &= \max_{v \in S} f(G, \pi_{<v}, v) \\ &= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\} \end{aligned}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\geq$:

- Wähle $\pi \in P(S)$ so, dass $F_G = \max_{v \in S} f(G, \pi_{<v}, v)$
- Sei w der letzte Knoten in der Reihenfolge, also $\pi(w) = |S|$, dann ist $\pi_{<w} = S \setminus \{w\}$

$$F_G(S) = \max_{v \in S} f(G, \pi_{<v}, v)$$
$$= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\}$$

$$F_G(S \setminus \{w\}) = \min_{\pi' \in P(S \setminus \{w\})} \max_{v \in S \setminus \{w\}} f(G, \pi'_{<v}, v) \leq \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\geq$:

- Wähle $\pi \in P(S)$ so, dass $F_G = \max_{v \in S} f(G, \pi_{<v}, v)$
- Sei w der letzte Knoten in der Reihenfolge, also $\pi(w) = |S|$, dann ist $\pi_{<w} = S \setminus \{w\}$

$$\begin{aligned} F_G(S) &= \max_{v \in S} f(G, \pi_{<v}, v) \\ &= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\} \\ &\geq \max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\} \end{aligned}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Karlsruher Institut für Technologie

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\leq$:

- Wähle $w \in S$ so, dass $\max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\}$ minimal ist
- Wähle $\pi' \in P(S \setminus \{w\})$ so, dass $F_G(S \setminus \{w\}) = \max_{v \in S \setminus \{w\}} f(G, \pi'_{<v}, v)$
- Sei π die Reihenfolge, die mit π' anfängt und mit w endet

$$\pi(v) = \begin{cases} \pi'(v) & v \neq w \\ |S| & v = w \end{cases}$$

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\leq$:

$$F_G(S) \leq \max_{v \in S} f(G, \pi_{<v}, v)$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \backslash \{v\}, v),$$
$$F_G(S \backslash \{v\})\}$$

$\leq$:

$$F_G(S) \leq \max_{v \in S} f(G, \pi_{<v}, v)$$
$$= \max\{f(G, \pi_{<w}, w), \max_{v \in S \backslash \{w\}} f(G, \pi_{<v}, v)\}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\leq$:

$$F_G(S) \leq \max_{v \in S} f(G, \pi_{<v}, v)$$
$$= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\}$$
$$= \max\{f(G, S \setminus \{w\}, w), \max_{v \in S \setminus \{w\}} f(G, \pi'_{<v}, v)\}$$

$$\pi(v) = \begin{cases} \pi'(v) & v \neq w \\ |S| & v = w \end{cases}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\varnothing) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\leq$:

$$\begin{aligned}
F_G(S) &\leq \max_{v \in S} f(G, \pi_{<v}, v) \\
&= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\} \\
&= \max\{f(G, S \setminus \{w\}, w), \max_{v \in S \setminus \{w\}} f(G, \pi'_{<v}, v)\} \\
&= \max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\}
\end{aligned}$$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beweis

Definition

$$F_G(S) = \min_{\pi \in P(S)} \max_{v \in S} f(G, \pi_{<v}, v),$$
$$F_G(\emptyset) = -\infty$$

$$F_G(S) = \min_{v \in S} \max\{f(G, S \setminus \{v\}, v), F_G(S \setminus \{v\})\}$$

$\leq$:

$$\begin{aligned}
F_G(S) &\leq \max_{v \in S} f(G, \pi_{<v}, v) \\
&= \max\{f(G, \pi_{<w}, w), \max_{v \in S \setminus \{w\}} f(G, \pi_{<v}, v)\} \\
&= \max\{f(G, S \setminus \{w\}, w), \max_{v \in S \setminus \{w\}} f(G, \pi'_{<v}, v)\} \\
&= \max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\} \\
&= \min_{w \in S} \max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\}
\end{aligned}$$

Pseudocode

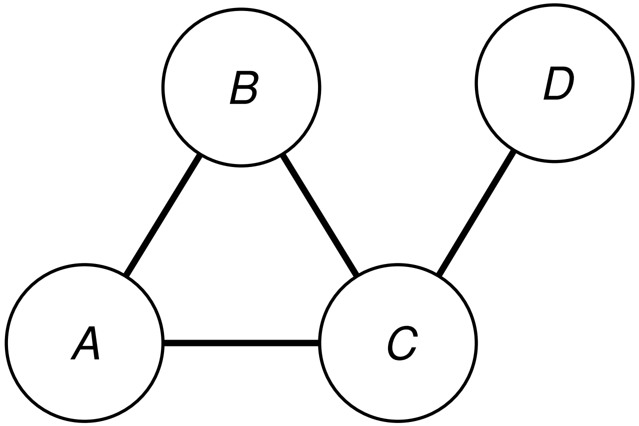
Algorithm 1: Dynamic-Programming-Algorithm ($G = (V, E)$)

$F_G(\emptyset) \coloneqq -\infty$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $F_G(S) \coloneqq \min_{w \in S} \max\{f(G, S \setminus \{w\}, w), F_G(S \setminus \{w\})\}$
  end for
end for
return $F_G(V)$

- Laufzeit: $O^*(2^n)$
- Speicherverbrauch: $O^*(2^n)$

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beispiel



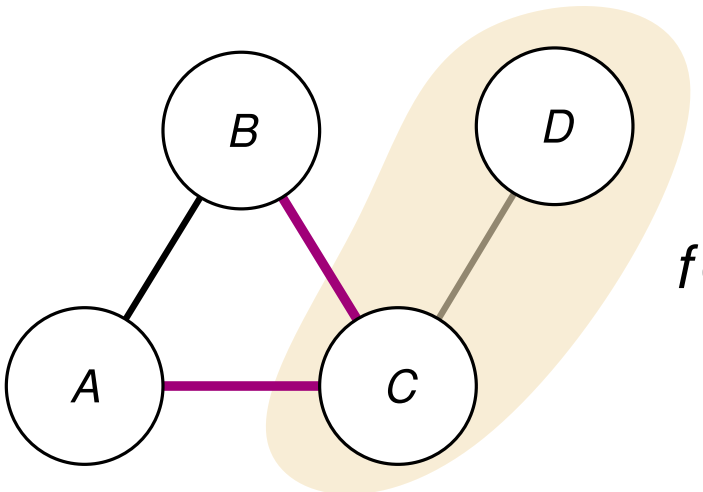
Optimal Linear Arrangement

Minimiere $\sum_{\{x,y\}\in E} |\pi(x) - \pi(y)| = \sum_{v\in V} |\{\{x,y\} \in E | \pi(x) \leq \pi(v) < \pi(y)\}|$,

also:

$f(G, L, v) = |\{\{x,y\} \in E | x \in L \cup \{v\}, y \in V \setminus (L \cup \{v\})\}|$

Beispiel



$f(G, \{C\}, D) = 2$

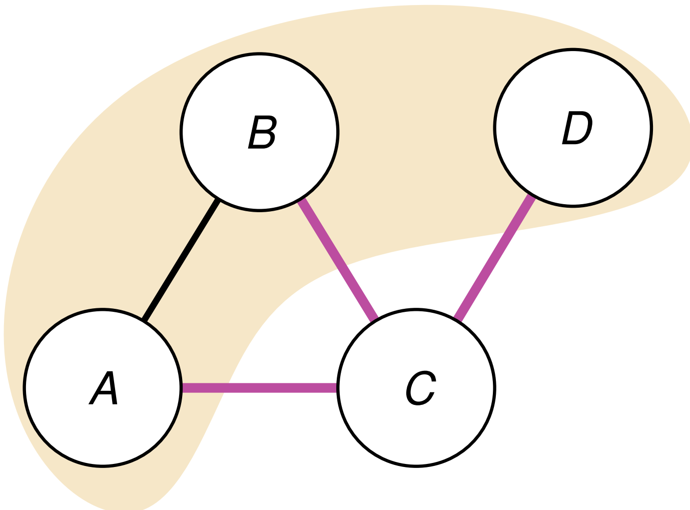
Optimal Linear Arrangement

Minimiere $\sum_{\{x,y\}\in E} |\pi(x) - \pi(y)| = \sum_{v \in V} |\{\{x,y\} \in E | \pi(x) \leq \pi(v) < \pi(y)\}|,$

also:

$f(G, L, v) = |\{\{x,y\} \in E | x \in L \cup \{v\}, y \in V \backslash (L \cup \{v\})\}|$

Beispiel



$f(G, \{A, D\}, B) = 3$

Optimal Linear Arrangement

Minimiere $\sum_{\{x,y\}\in E} |\pi(x) - \pi(y)| = \sum_{v \in V} |\{\{x, y\} \in E | \pi(x) \leq \pi(v) < \pi(y)\}|$,

also:

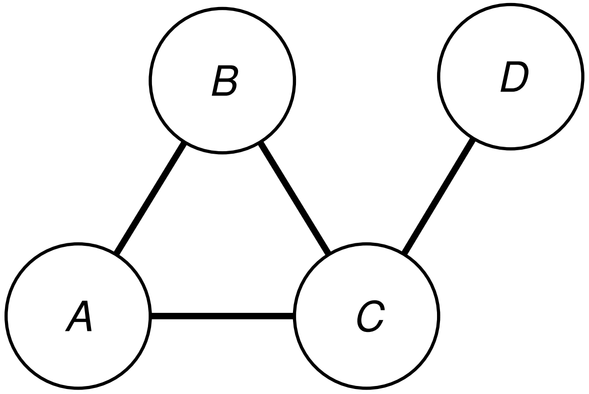
$f(G, L, v) = |\{\{x, y\} \in E | x \in L \cup \{v\}, y \in V \setminus (L \cup \{v\})\}|$

Beispiel

- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

$F(\emptyset) := 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beispiel

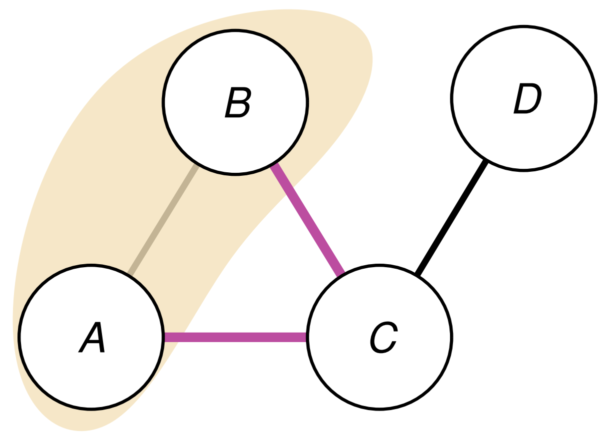
- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

$F(\emptyset) \coloneqq 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) \coloneqq \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$

- $i = 2$

S	w	$f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
$\{A, B\}$	B	$f(\{A\}, B) + F(\{A\})$
	A	$f(\{B\}, A) + F(\{B\})$



Beispiel

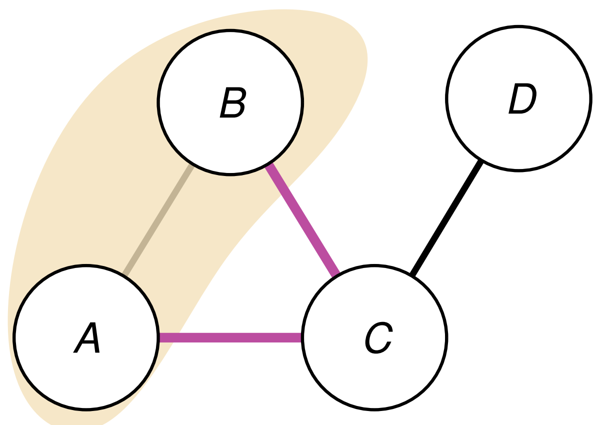
- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

$F(\emptyset) :=\ 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$

- $i = 2$

S	w	$f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
$\{A, B\}$	B	4
	A	$f(\{B\}, A) + F(\{B\})$



Beispiel

- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

$F(\emptyset) := 0$

for $i = 1..n$ **do**

  for *each* $S \subseteq V, |S| = i$ **do**

    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$

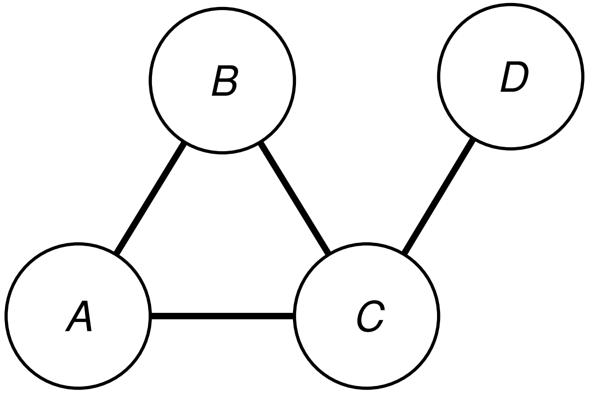
  end for

end for

return $F(V)$

- $i = 2$

S	w	$f(S\setminus\{w\}, w) + F(S\setminus\{w\})$
$\{A, B\}$	B	4
	A	4



Beispiel

Karlsruher Institut für Technologie

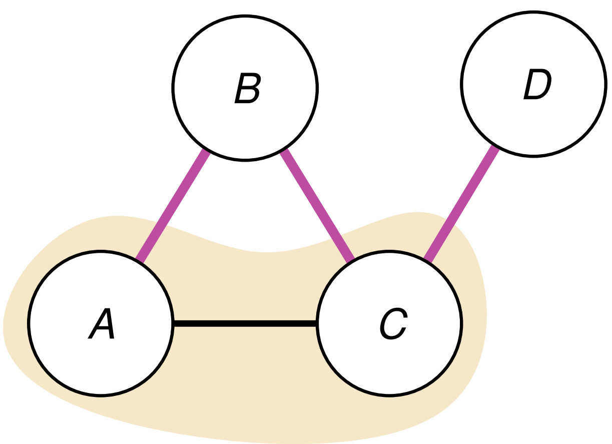
- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

$F(\emptyset) :=\ 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$

- $i = 2$

S	$F(S)$		
$\{A, B\}$	4		
$\{A, C\}$		w	$f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
		C	5
		A	6



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Beispiel

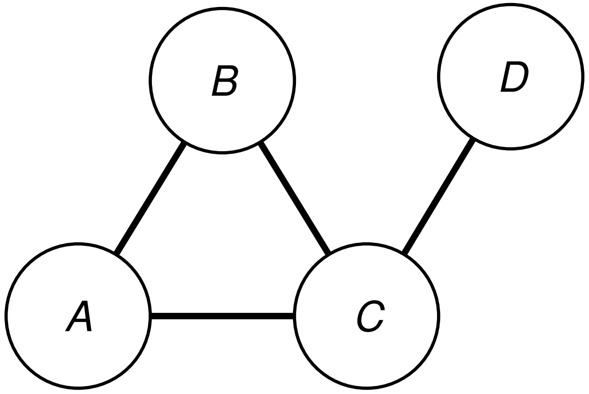
- $i = 1$

S	$F(S)$
$\{A\}$	2
$\{B\}$	2
$\{C\}$	3
$\{D\}$	1

- $i = 2$

S	$F(S)$
$\{A, B\}$	4
$\{A, C\}$	5
$\{A, D\}$	5
$\{B, C\}$	5
$\{B, D\}$	4
$\{C, D\}$	3

$F(\emptyset) := 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$

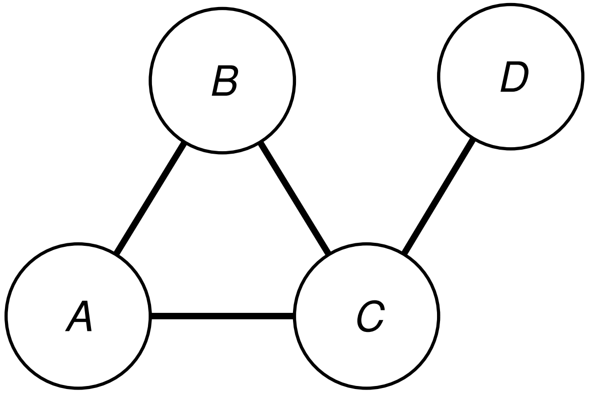


Beispiel

- $i = 3$

S	$F(S)$
$\{A, B, C\}$	5
$\{A, B, D\}$	7
$\{A, C, D\}$	5
$\{B, C, D\}$	5

> $F(\emptyset) := 0$
> **for** $i = 1..n$ **do**
>   **for** *each* $S \subseteq V, |S| = i$ **do**
> $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
>   **end for**
> **end for**
> **return** $F(V)$



Beispiel

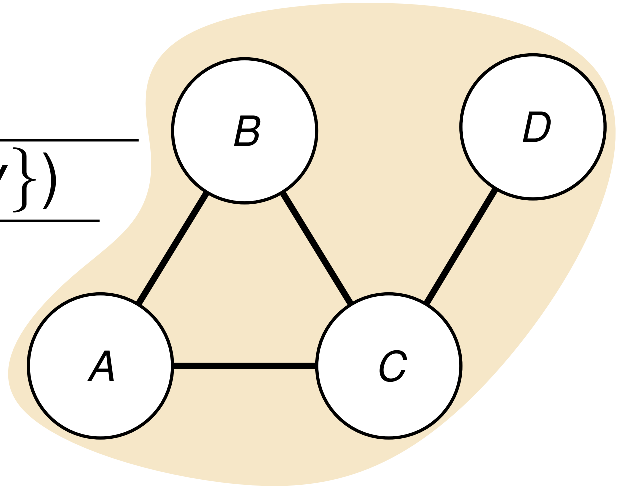
- $i = 3$

S	$F(S)$
$\{A, B, C\}$	5
$\{A, B, D\}$	7
$\{A, C, D\}$	5
$\{B, C, D\}$	5

$F(\emptyset) := 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$

- $i = 4$

S	w	$f(S\setminus\{w\}, w) + F(S\setminus\{w\})$
$\{A, B, C, D\}$	D	5
	C	7
	B	5
	A	5



Beispiel

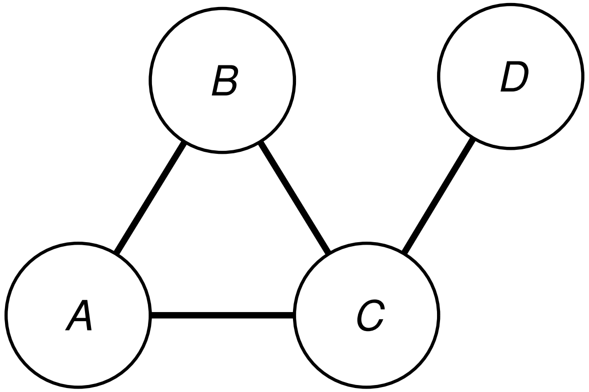
- $i = 3$

S	$F(S)$
$\{A, B, C\}$	5
$\{A, B, D\}$	7
$\{A, C, D\}$	5
$\{B, C, D\}$	5

- $i = 4$

S	$F(S)$
$\{A, B, C, D\}$	5

$F(\emptyset) := 0$
for $i = 1..n$ **do**
  for *each* $S \subseteq V, |S| = i$ **do**
    $A(S) := \min_{w \in S} f(S \setminus \{w\}, w) + F(S \setminus \{w\})$
  end for
end for
return $F(V)$



2. Algorithmus
Rekursion

Pseudocode

Algorithm 2: Recursive(Graph G, vertex set L, vertex set R)

```
if R = {v} then
    return f(G, L, v)
end if
opt := ∞
for each R' ⊆ R, |R'| = ⌊|R|/2⌋ do
    v1 := Recursive(G, L, R')
    v2 := Recursive(G, L ∪ R', R \ R')
    opt := min{opt, max{v1, v2}}
end for
return opt
```

- Laufzeit: $O^*(4^n)$
- Speicherverbrauch: polynomiell

Institut für Theoretische Informatik
Lehrstuhl Algorithmik

Laufzeit

Laufzeit: $O^*(4^n)$:

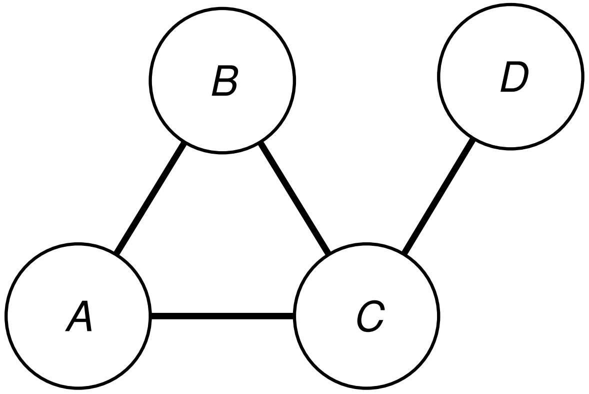
- Rekursive Aufrufe:

$$T(k) \leq \binom{k}{\left\lfloor \frac{k}{2} \right\rfloor} \left(T\left(\left\lfloor \frac{k}{2} \right\rfloor \right) + T\left(\left\lceil \frac{k}{2} \right\rceil \right) + 2 \right)$$

- Es folgt: $T(k) \leq 4^k$

$\text{opt} := \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$\quad v_1 := \text{Recursive}(L, R')$
$\quad v_2 := \text{Recursive}(L \cup R', R \setminus R')$
$\quad \text{opt} := \min\{\text{opt}, v_1 + v_2\}$
end for

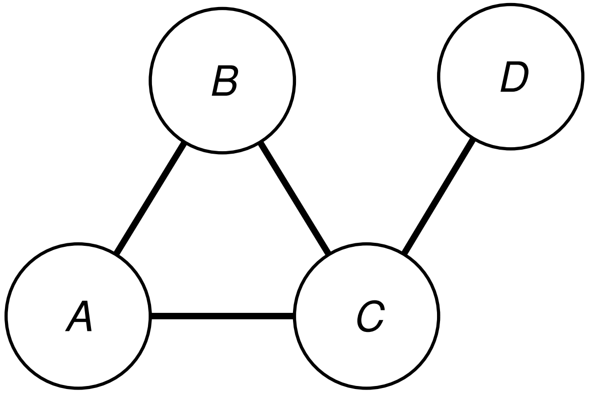
$\text{Recursive}(\{\}, \{A, B, C, D\})$:



$$\text{Recursive}(\{\}, \{A, B, C, D\})\text{: opt} = \infty$$

$$R' = \{A, B\}$$

opt $:= \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
  $v_1 :=$ Recursive(L, R')
  $v_2 :=$ Recursive$(L \cup R', R \setminus R')$
  opt $:= \min\{\text{opt}, v_1 + v_2\}$
end for

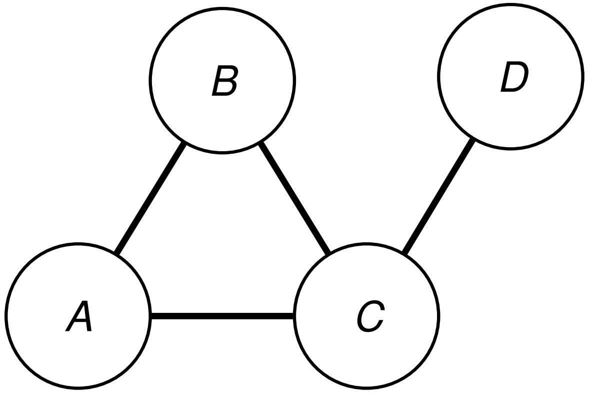


$$\begin{array}{l}
\text{opt} := \infty \\
\textbf{for } \textit{each } R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor \textbf{ do} \\
\quad v_1 := \text{Recursive}(L, R') \\
\quad v_2 := \text{Recursive}(L \cup R', R \setminus R') \\
\quad \text{opt} := \min\{\text{opt}, v_1 + v_2\} \\
\textbf{end for}
\end{array}$$

Recursive($\{\}, \{A, B, C, D\}$): opt $= \infty$

$R' = \{A, B\}$

$v_1 = $ Recursive($\{\}, \{A, B\}$):



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

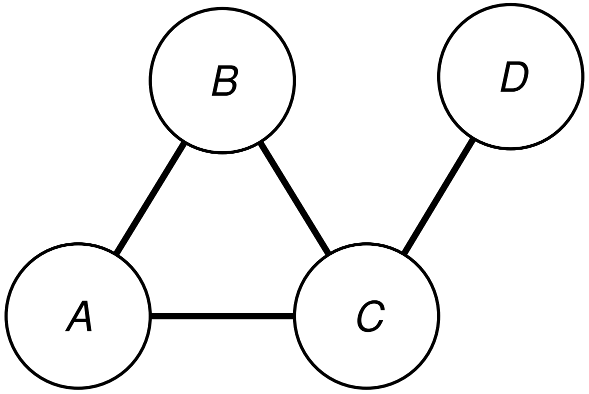
$$
\begin{aligned}
&\text{opt} \coloneqq \infty \\
&\textbf{for } \textit{each } R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor \textbf{ do} \\
&\quad v_1 \coloneqq \text{Recursive}(L, R') \\
&\quad v_2 \coloneqq \text{Recursive}(L \cup R', R \setminus R') \\
&\quad \text{opt} \coloneqq \min\{\text{opt}, v_1 + v_2\} \\
&\textbf{end for}
\end{aligned}
$$

Recursive($\{\}, \{A, B, C, D\}$): opt $= \infty$

$R' = \{A, B\}$

$v_1 =$ Recursive($\{\}, \{A, B\}$): opt $= \infty$

$R' = \{A\}$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

$$
\begin{array}{l}
\text{opt} := \infty \\
\textbf{for } \textit{each } R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor \textbf{ do} \\
\quad v_1 := \text{Recursive}(L, R') \\
\quad v_2 := \text{Recursive}(L \cup R', R \setminus R') \\
\quad \text{opt} := \min\{\text{opt}, v_1 + v_2\} \\
\textbf{end for}
\end{array}
$$

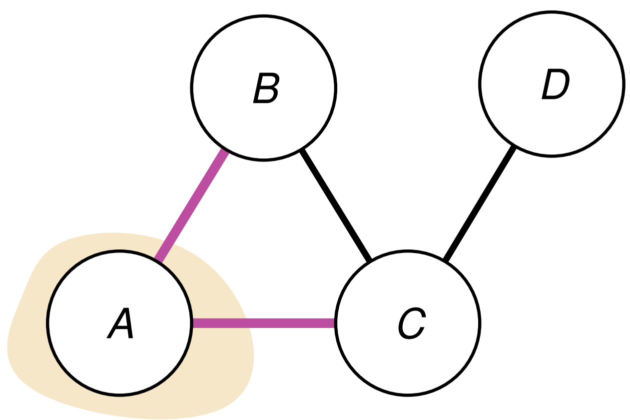
Recursive($\{\}, \{A, B, C, D\}$): opt $= \infty$

$R' = \{A, B\}$

$v_1 =$ Recursive($\{\}, \{A, B\}$): opt $= \infty$

$R' = \{A\}$

$v_1 =$ Recursive($\{\}, \{A\}$):



$$
\begin{aligned}
&\text{opt} \coloneqq \infty \\
&\textbf{for } \textit{each } R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor \textbf{ do} \\
&\quad v_1 \coloneqq \text{Recursive}(L, R') \\
&\quad v_2 \coloneqq \text{Recursive}(L \cup R', R \setminus R') \\
&\quad \text{opt} \coloneqq \min\{\text{opt}, v_1 + v_2\} \\
&\textbf{end for}
\end{aligned}
$$

Recursive($\{\}, \{A, B, C, D\}$): opt $= \infty$

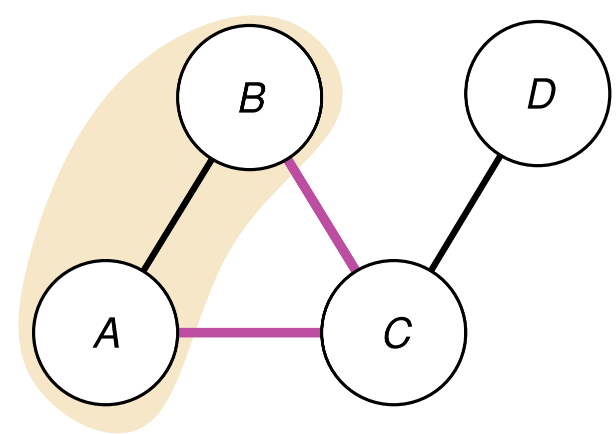
$R' = \{A, B\}$

$v_1 =$ Recursive($\{\}, \{A, B\}$): opt $= \infty$

$R' = \{A\}$

$v_1 =$ Recursive($\{\}, \{A\}$): 2

$v_2 =$ Recursive($\{A\}, \{B\}$):



$\mathrm{opt} := \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
  $v_1 := \mathrm{Recursive}(L, R')$
  $v_2 := \mathrm{Recursive}(L \cup R', R \setminus R')$
  $\mathrm{opt} := \min\{\mathrm{opt}, v_1 + v_2\}$
end for

$\mathrm{Recursive}(\{\}, \{A, B, C, D\})$: $\mathrm{opt} = \infty$

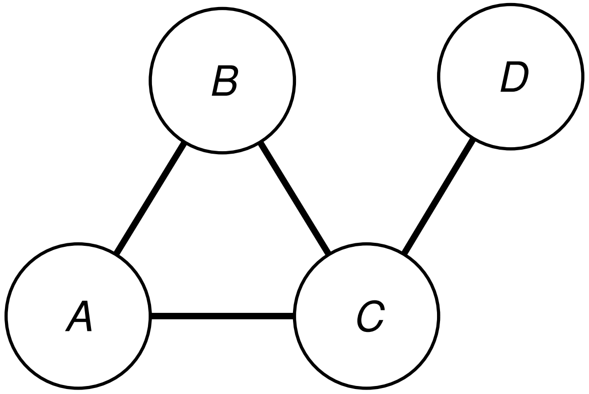
$R' = \{A, B\}$

$v_1 = \mathrm{Recursive}(\{\}, \{A, B\})$: $\mathrm{opt} = \infty$

$R' = \{A\}$ $\quad v_1 + v_2 = 4$

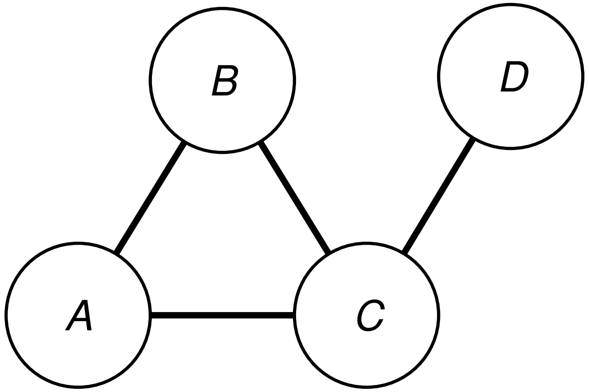
$v_1 = \mathrm{Recursive}(\{\}, \{A\})$: 2

$v_2 = \mathrm{Recursive}(\{A\}, \{B\})$: 2



$\text{opt} \coloneqq \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$\quad v_1 \coloneqq \text{Recursive}(L, R')$
$\quad v_2 \coloneqq \text{Recursive}(L \cup R', R \backslash R')$
$\quad \text{opt} \coloneqq \min\{\text{opt}, v_1 + v_2\}$
end for

$\text{Recursive}(\{\}, \{A, B, C, D\})$: $\text{opt} = \infty$
$\quad R' = \{A, B\}$
$\quad\quad v_1 = \text{Recursive}(\{\}, \{A, B\})$: $\text{opt} = 4$
$\quad\quad\quad R' = \{B\}$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

$\text{opt} \coloneqq \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$v_1 \coloneqq \text{Recursive}(L, R')$
$v_2 \coloneqq \text{Recursive}(L \cup R', R \setminus R')$
$\text{opt} \coloneqq \min\{\text{opt}, v_1 + v_2\}$
end for

$\text{Recursive}(\{\}, \{A, B, C, D\})$: $\text{opt} = \infty$

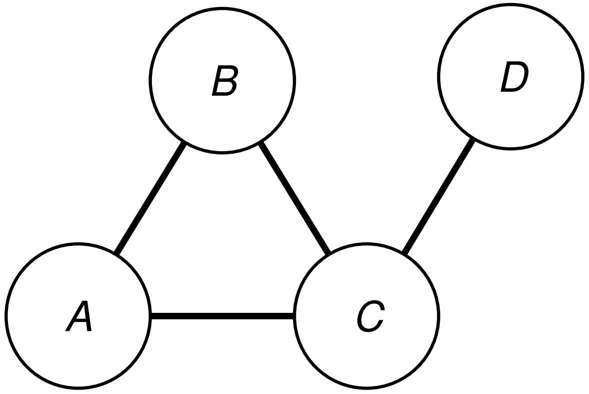
$R' = \{A, B\}$

$v_1 = \text{Recursive}(\{\}, \{A, B\})$: $\text{opt} = 4$

$R' = \{B\}$ $\quad v_1 + v_2 = 4$

$v_1 = \text{Recursive}(\{\}, \{B\})$: 2

$v_2 = \text{Recursive}(\{B\}, \{A\})$: 2



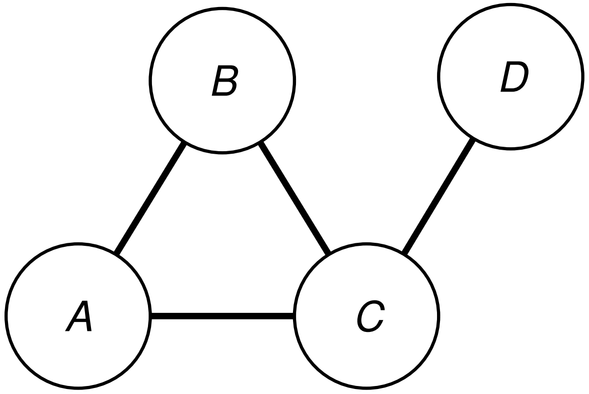
$\text{opt} \coloneqq \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$v_1 \coloneqq \text{Recursive}(L, R')$
$v_2 \coloneqq \text{Recursive}(L \cup R', R \setminus R')$
$\text{opt} \coloneqq \min\{\text{opt}, v_1 + v_2\}$
end for

$\text{Recursive}(\{\}, \{A, B, C, D\})\text{: opt} = \infty$

$R' = \{A, B\}$

$v_1 = \text{Recursive}(\{\}, \{A, B\})\text{: } 4$

$v_2 = \text{Recursive}(\{A, B\}, \{C, D\})\text{:}$



Institut für Theoretische Informatik
Lehrstuhl Algorithmik

$\text{opt} := \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$v_1 := \text{Recursive}(L, R')$
$v_2 := \text{Recursive}(L \cup R', R \setminus R')$
$\text{opt} := \min\{\text{opt}, v_1 + v_2\}$
end for

$\text{Recursive}(\{\}, \{A, B, C, D\})$: $\text{opt} = \infty$

$R' = \{A, B\}$

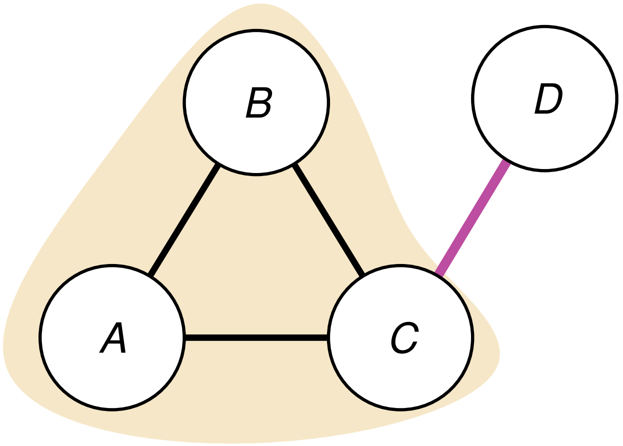
$v_1 = \text{Recursive}(\{\}, \{A, B\})$: 4

$v_2 = \text{Recursive}(\{A, B\}, \{C, D\})$: $\text{opt} = \infty$

$R' = \{C\}$ $\quad v_1 + v_2 = 1$

$v_1 = \text{Recursive}(\{A, B\}, \{C\})$: 1

$v_2 = \text{Recursive}(\{A, B, C\}, \{D\})$: 0



$$
\begin{array}{l}
\text{opt} := \infty \\
\textbf{for } \textit{each } R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor \textbf{ do} \\
\quad v_1 := \text{Recursive}(L, R') \\
\quad v_2 := \text{Recursive}(L \cup R', R \setminus R') \\
\quad \text{opt} := \min\{\text{opt}, v_1 + v_2\} \\
\textbf{end for}
\end{array}
$$

$\text{Recursive}(\{\}, \{A, B, C, D\})$: opt $= \infty$

$R' = \{A, B\}$

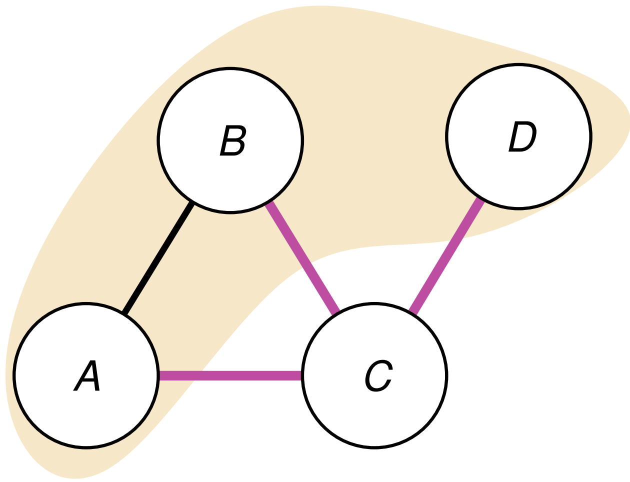
$v_1 = \text{Recursive}(\{\}, \{A, B\})$: 4

$v_2 = \text{Recursive}(\{A, B\}, \{C, D\})$: opt $= 1$

$R' = \{D\}$ $\qquad v_1 + v_2 = 3$

$v_1 = \text{Recursive}(\{A, B\}, \{D\})$: 3

$v_2 = \text{Recursive}(\{A, B, D\}, \{C\})$: 0



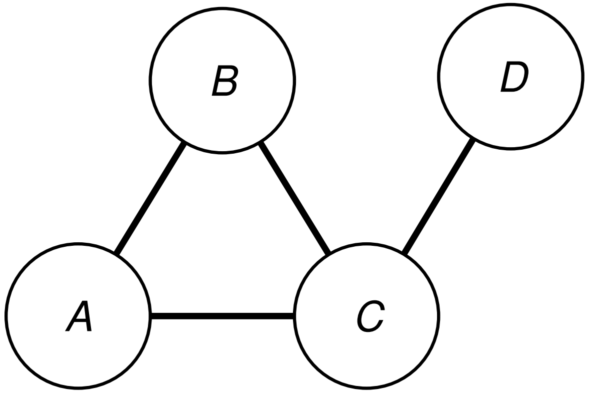
```
opt := ∞
for each R' ⊆ R, |R'| = ⌊|R|/2⌋ do
    v_1 := Recursive(L, R')
    v_2 := Recursive(L ∪ R', R\R')
    opt := min{opt, v_1 + v_2}
end for
```

Recursive($\{\}, \{A, B, C, D\}$): opt $= \infty$

$R' = \{A, B\}$ $\qquad v_1 + v_2 = 5$

$v_1 =$ Recursive($\{\}, \{A, B\}$): 4

$v_2 =$ Recursive($\{A, B\}, \{C, D\}$): 1



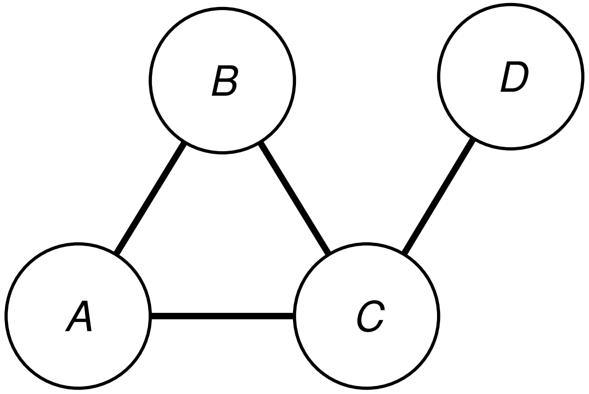
$$\text{Recursive}(\{\}, \{A, B, C, D\})\text{: opt} = 5$$

$R' = \{A, B\}$ $\qquad v_1 + v_2 = 5$

$v_1 = \text{Recursive}(\{\}, \{A, B\})\text{: } 4$

$v_2 = \text{Recursive}(\{A, B\}, \{C, D\})\text{: } 1$

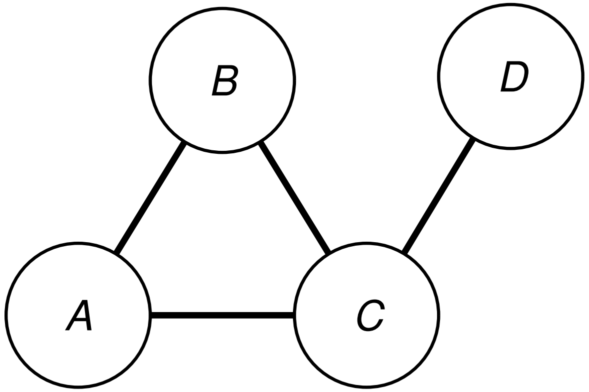
opt $:= \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
    $v_1 := \text{Recursive}(L, R')$
    $v_2 := \text{Recursive}(L \cup R', R \setminus R')$
    opt $:= \min\{\text{opt}, v_1 + v_2\}$
end for



$\text{opt} := \infty$
for *each* $R' \subseteq R, |R'| = \lfloor |R|/2 \rfloor$ **do**
$v_1 := \text{Recursive}(L, R')$
$v_2 := \text{Recursive}(L \cup R', R \setminus R')$
$\text{opt} := \min\{\text{opt}, v_1 + v_2\}$
end for

$\text{Recursive}(\{\}, \{A, B, C, D\})$: $\text{opt} = 5$

$R' = \{A, C\}$



Abschließendes

Karlsruher Institut für Technologie

- 4^n wächst schnell
- verschiedene interessante Probleme können als Vertex Ordering Probleme formuliert werden:
  TREEWIDTH, MINIMUM FILL-IN, PATHWIDTH, SUM CUT, MINIMUM INTERVAL GRAPH COMPLETION, CUTWIDTH, OPTIMAL LINEAR ARRANGEMENT, DIRECTED FEEDBACK ARC SET

Danke fürs Zuhören