Bsp (Unit-Time-Tasks)

Aufg.	Schlusstermin	Gewinn
1	2	30
2	1	35
3	2	25
4	1	40

Anordnung (1,2) wäre nicht zulässig

(3,1) zulässig (Gewinn: 25+30=55)

Optimal?

↳ Greedy: 1 Sortieren nach fallenden Gewinn

2 Hinzufügen, wenn Auswahl zulässig bleibt

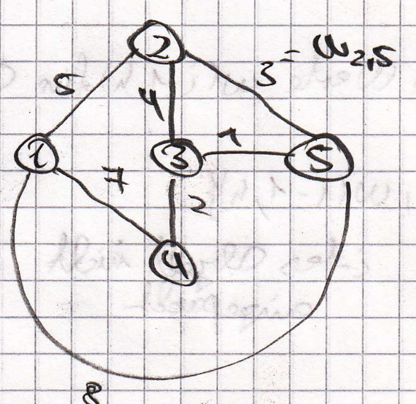
$I=\{4\}$, $I=\{4,2\}$ nicht zulässig

$I=\{4,1\}$ zulässig

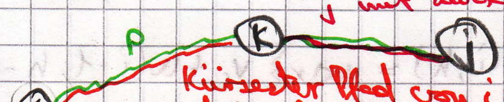
$I=\{4,3\}$ nicht zulässig

$\Rightarrow$ optimal (4,1) $\rightarrow$ Gewinn=70

Floyd-Warshall-Algorithmus



Optimale Teilstruktur?



Kürzester Pfad von k nach j mit Zwischenknoten aus $\{1,\dots k-1\}$

Kürzester Pfad von i nach k mit Zwischenknoten aus $\{1,\dots k-1\}$

p ist kürzester Pfad von i nach j mit Zwischenknoten aus $\{i\dots k\}$

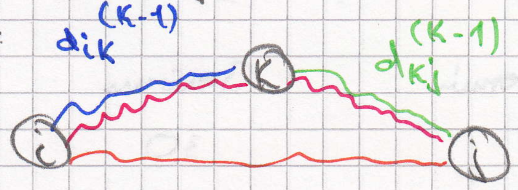
Gesucht ist $d_{ij}^{(n)}$ mit $1\leq j\leq n$

$1\leq i\leq n$

Annahme: $d_{ij}^{(k-1)}$ ist bekannt

Rekursion für $d_{ij}^{(K)}$

2 Fälle:



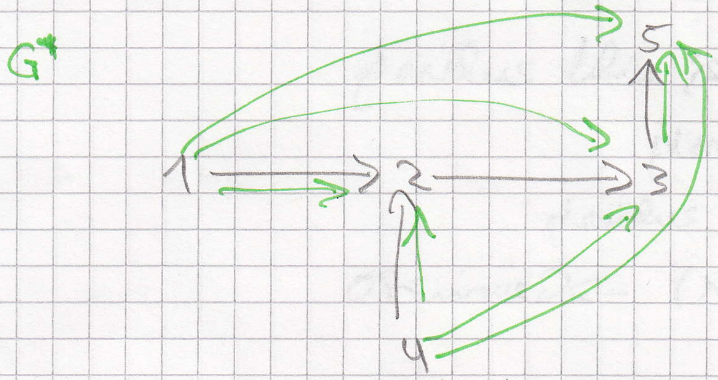
i) Kürzester Pfad von i nach j mit ZK aus $\{1 \dots K\}$ verläuft nicht über K $\quad d_{ij}^{(K)} = d_{ij}^{(K-1)}$

ii) kürzester Pfad von i nach j mit ZK aus $\{1, \dots K\}$ verläuft über K $\quad d_{ij}^{(K)} = d_{iK}^{(K-1)} + d_{Kj}^{(K-1)}$

Insgesamt

$$d_{ij}^{(K)} = \min\{d_{ij}^{(K-1)}, d_{iK}^{(K-1)} + d_{Kj}^{(K-1)}\}$$

Bsp: Transitive Hülle eines Graphen



$$t_{ij}^{(K)} = t_{ij}^{(K-1)} \underset{(\text{oder})}{\vee} (t_{iK}^{(K-1)} \underset{(\text{und})}{\wedge} t_{Kj}^{(K-1)})$$

0-1-Rucksack-Problem

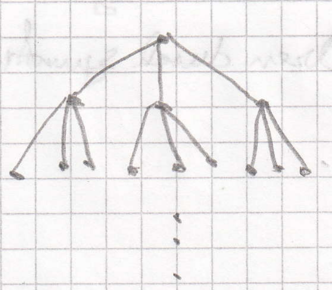
Alle Werte $w(i-1,h)$ für $0 \leq h \leq G$ sind bekannt

$$w(i,h) = \max\{\underbrace{v_i + w(i-1, h-g_i)}_{\text{i-tes Objekt wird eingepackt}}, \underbrace{w(i-1,h)}_{\text{i-tes Objekt nicht eingepackt}}\}$$

Zur minimalen Höhe eines d-Knoten Baum

Maximaler Baum: hat bei gegebener Höhe h die maximale Knoten-Anzahl N(h)

z.B. $d=3$



3^0 - Knoten

3^1 - Knoten

3^2 - Knoten

3^h - Knoten

$$N(h) = \sum_{k=0}^{h} d^k = 1 + d^1 + d^2 + d^3 + \dots + d^h = \frac{d^{h+1} - 1}{d-1}$$ geometrische Summe

Vollständiger Baum: alle Ebenen bis auf die letzte sind vollständig besetzt.

Hat unter allen Bäumen mit n Knoten die minimale Höhe h

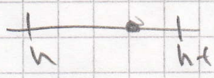
$$N(h-1) < n \le N(h)$$

$$\Rightarrow \quad \frac{d^h - 1}{d-1} < n \le \frac{d^{h+1} - 1}{d-1}$$

$$d^h - 1 < n(d-1) \le d^{h+1} - 1$$

$$d^h < n(d-1) + 1 \le d^{h+1}$$

$$\underset{\in \mathbb{N}}{h} < \underbrace{\log_d (n(d-1)+1)}_{\in \mathbb{R}} \le \underset{\in \mathbb{N}}{h+1}$$

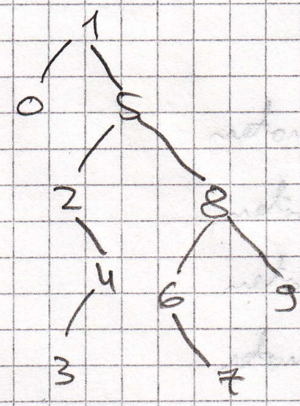


$$\Rightarrow h+1 = \lceil \log_d (n(d-1)+1) \rceil$$

$$\Rightarrow h = \lceil \log_d (n(d-1)+1) \rceil - 1$$

Binäre Suchbäume

Bsp: 1, 0, 5, 2, 8, 4, 6, 9, 3, 7

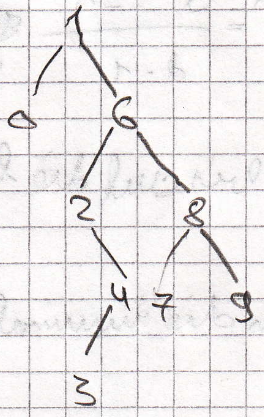


Löschen von 5:

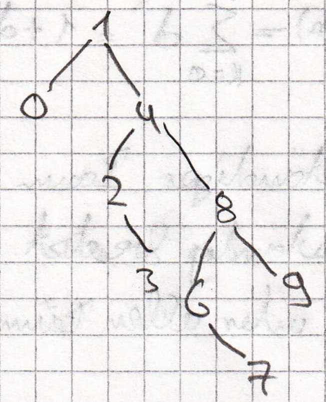
i) Ersetzen durch symmetr. Nachfolger

ii) Ersetzen durch symmetr. Vorgänger

i)



ii)

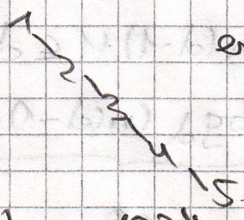


In-order-Durchlauf: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

= aufsteigend sortierte Folge

Aufwand Wörterbuchoperationen: $O(h)$

Bsp: 1, 2, 3, 4, 5



entartet zu Liste

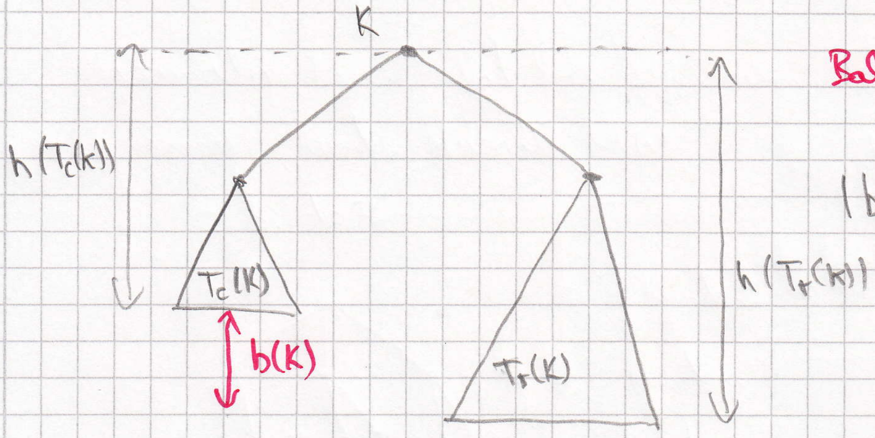
$O(h) = O(n)$

Schlecht

gut: $h = O(\log_2(n))$ $n = 1024$

Balanciert

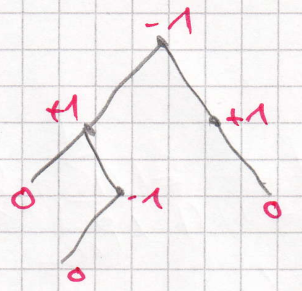
AVL-Bäume



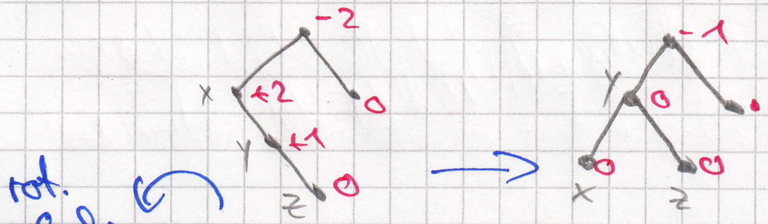
Balance $b(k) = h(T_r(k)) - h(T_l(k))$

$|b(k)| \leq 1$ AVL-Bedingung

Bsp



AVL-Bed. erfüllt



AVL-Bed nicht erfüllt

Bsp. Einfügen in AVL-Baum

4, 5, 7, 2, 1, 3, 6 in leeren Baum