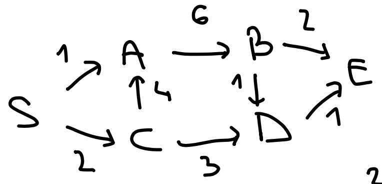
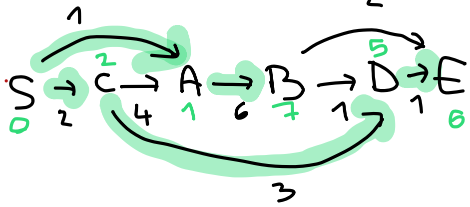
6. Dinamično programiranje

6.1. Najkrajše poti v DAGih





- če gre najcenejša pot $s \to \cdots \to v$ čez u, potem je $s \to \cdots \to u$ najcenejša pot do u.
- razdalja[v] je odvisna le od vrednosti razdalja[u], kjer je u vozlišče, ki v topološki ureditvi nastopa pred v.

razdalja[s] = 0

za vse $v \in V$ v topološki ureditvi:

$$\text{razdalja}[v] = \min_{u \to v \in E} \{ l_{u,v} + \text{razdalja}[u] \}$$

- enak postopek bi lahko uporabili, če bi iskali najdražjo pot, če bi nas namesto vsote cen zanimal produkt, ali kakršna koli druga funkcija.

Splošnemu pristopu, ki sledi načelu, da so rešitve odvisne od rešitev predhodnikov, pravimo dinamično programiranje.

Bellman

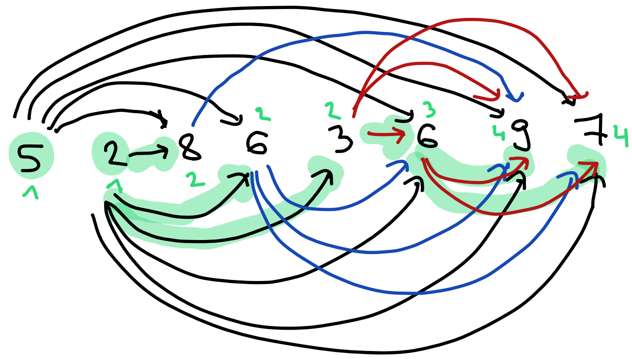
*1920 NY, NY, ZDA

†1984 LA, CA, ZDA

6.2. Najdaljše naraščajoče podzaporedje

Imejmo končno zaporedje $a_1, a_2, \dots, a_n$.
Iščemo najdaljše zaporedje indeksov $1 \le i_1 < i_2 < \dots < i_k \le n$,
da je $a_{i_1} \le a_{i_2} \le \dots \le a_{i_k}$

5 2 8 6 3 6 9 7



vozlišča ... indeksi $1 \dots n$

oznaka vozlišča ... najdaljše podzap. $a_1, \dots, a_i$

povezave ... kdaj se rešitev enega podproblema lahko pojavi v drugem

$V = \{1, \dots, n\}$
$E = \{(i,j) \mid 1 \le i < j \le n,\ a_i \le a_j\}$

za vse $v \in V$ v topološki ureditvi:

$$dolzina[v] = \max_{u \to v \in E} \{1 + dolzina[u]\}$$

Če povezav $u \to v \in E$ ni, nastavimo $dolzina[v] = 1$

6.3. Urejevalna (Levenshteinova) razdalja

D.N.

6.4. Problem nahrbtnika

Imamo predmete z vrednostmi $v_1, \dots, v_n$ ter težami $w_1, \dots, w_n$.
Koliko katerih naj vzamemo, da bo skupna teža $\le W$,
kapaciteta
skupna vrednost pa čim večja.

i	v_i	w_i	$\rho_i = v_i / w_i$
1	30	6	5
2	14	3	4,67
3	16	4	4
4	9	2	4,5

Tobias Dantzig ~~George~~

$W = 10$

		neomejeno kopij DA	NE
režemo	DA	1	2
	NE	3	4

1\. Poiščemo predmet z največjo gostoto ρ_1 ter vzamemo $\frac{W}{w_1}$ kosov.

režemo neomejeno

Vrednost bo $\frac{W}{w_1} \cdot v_1 = W \cdot \rho_1$.

Čas. zahtevnost: $O(n)$

2\. Uredimo po gostoti in po vrsti cele dajemo v nahrbtnik dokler lahko.

režemo omejeno

Zadnji predmet dodamo v deležu sorazmernim s preostalim prostorom. $O(n \cdot \log n)$

3\. Rešujemo z D.P.

ne režemo neomejeno

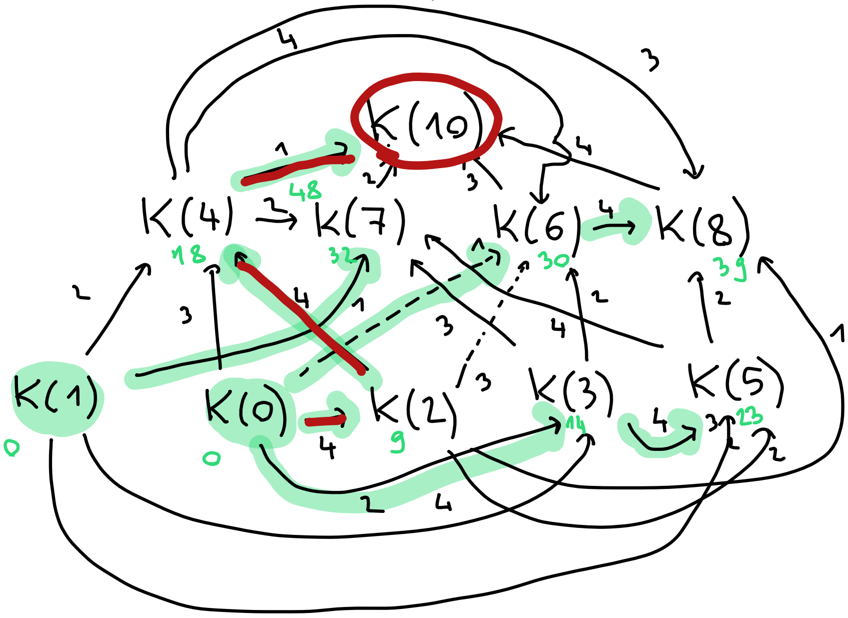
$K(w)$ ... optimalna napolnitev nahrbtnika s kapaciteto w

$K(0) = 0$ ← delamo samo z vrednostmi seznam predmetov = D.N.

$$K(w) = \max_{\substack{1 \le i \le n \\ w_i \le w}} \{ v_i + K(w - w_i) \}$$

Iščemo $K(W)$

i	v_i	w_i
1	30	6
2	14	3
3	16	4
4	9	2



$K(10) = v_1 + v_4 + v_4 = 48$

če je celoštevilsko

↓

$O(W \cdot n)$

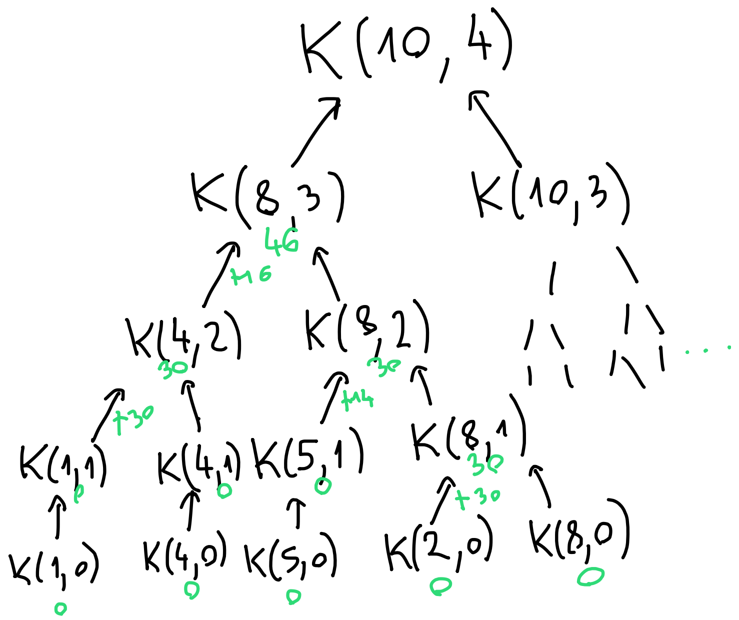
4.

ne režemo
omejeno kopij

$K(w, j)$ … optimalna napolnitev nahrbtnika s kapaciteto w, če se omejimo na predmete $1 \dots j$

$$K(w,j) = \max\{\underbrace{K(w-w_j, j-1)+v_j}_{\text{če } w_j \leq w}, K(w, j-1)\}$$

$$K(w,0) = 0$$



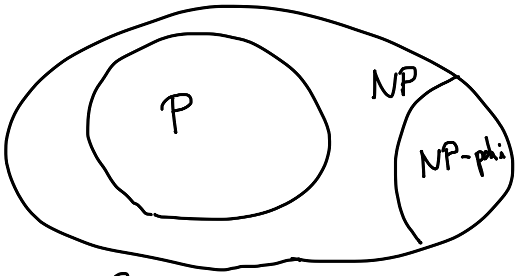
i	v_i	w_i
1	30	6
2	14	3
3	16	4
4	9	2

$O(2^n)$

7. Linearno programiranje - opt. metode
8. NP - polni problemi

P … problemi, ki jih znamo rešiti v polinomskem času (praktično vsi, ki smo jih videli do sedaj)

NP … problemi, za katere znamo v polinomskem času preveriti pravilnost dane rešitve



$P \stackrel{?}{=} NP$

max ust. napolnitev V^*

ali obstaja ust. nap z vrednostjo $\geq V$ DA / NE

$Q_1 \rightleftarrows Q_2$ TSP SAT

9. Hevristike za učinkovitejše reševanje NP-polnih problemov