Dinamično Programiranje

Memoizacija

```
def f(x):      } počasna
    ⋮
```

Ko izračunamo $f(x_0) \to y_0$, bi si zapomnili $x_0 \xmapsto{f} y_0$, da ne bomo naslednjič spet računali $f(x_0)$.

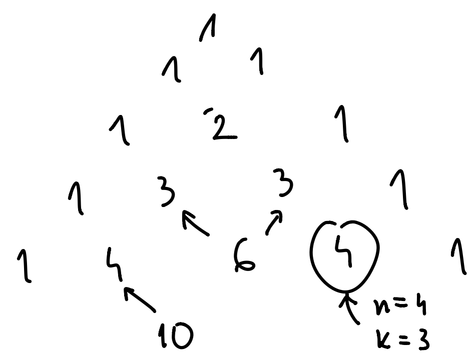
$F_0 = 0$

$F_1 = 1$

$F_n = F_{n-1} + F_{n-2}$

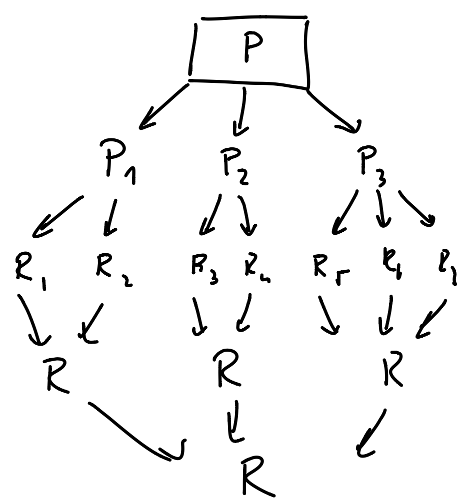
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

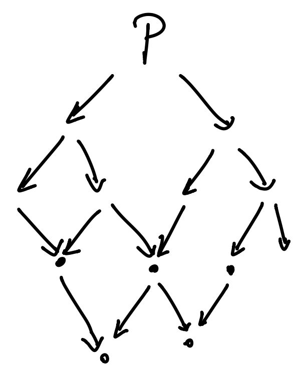
$C(n,k)$



Dinamično programiranje

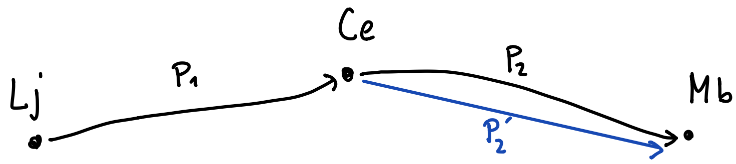
Deli & vladaj





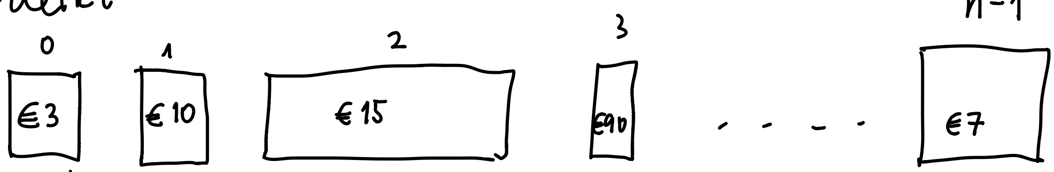
Če se podproblemi ponavljajo, jih memoiziramo!

Dinamično programiranje: (glej zapiske!)

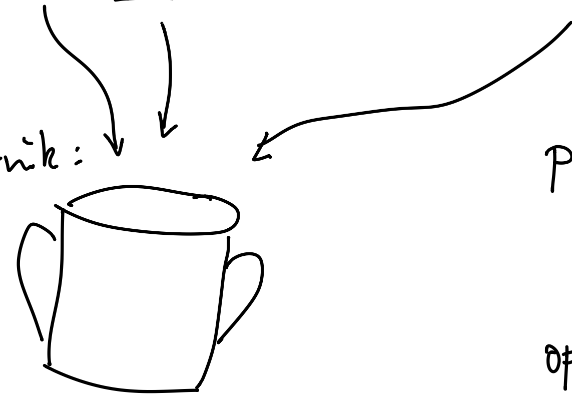


Problem 0-1 nahrbtnika

Izdelki



nahrbtnik:



↑To ni lonec

Pogoj: skupna velikost izdelkov ne presega velikosti nahrbtnika

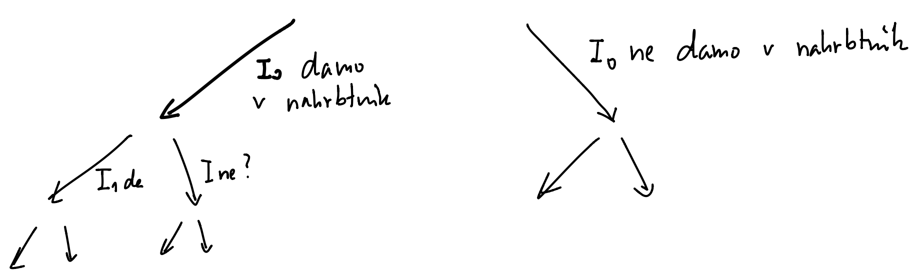
optimiziramo: skupna vrednost izdelkov

Naivna rešitev: preizkusimo vse možnosti → vse podmnožice izdelkov

2^n možnosti

Deli & vladaj:

$I_0, I_1, \ldots, I_{n-1}$



Optimalnost:

$N(i,k)$ problem 01-nahrbtnika z nahrbtnikom velikosti k in izdelki $I_i, I_{i+1}, \ldots, I_{n-1}$

$v(i,k)$ vrednost optimalne rešitve problema $N(i,k)$

$N(i,k)$

→ uporabimo I_i: podproblem $N(i+1, k - \text{velikost } I_i)$, vrednost $N(i+1, k - \text{velikost } I_i) +$ vrednost I_i

→ ne uporabimo I_i: podproblem $N(i+1, k)$, vrednost $N(i+1, k)$

$$N(i,k) = \max\left(N(i+1,k),\ N(i+1, k - \text{velikost } I_i) + \text{vrednost } I_i\right)$$

Ali se podproblemi prekrivajo?

$N(i,k)$ kjer je $0 \le i < n$ in $0 \le k \le$ velikost nahrbtnika K

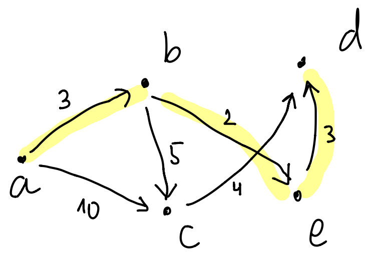
skupaj $O(n \cdot K)$ podproblemov

Da, če je $n \cdot K$ manjši od 2^n.

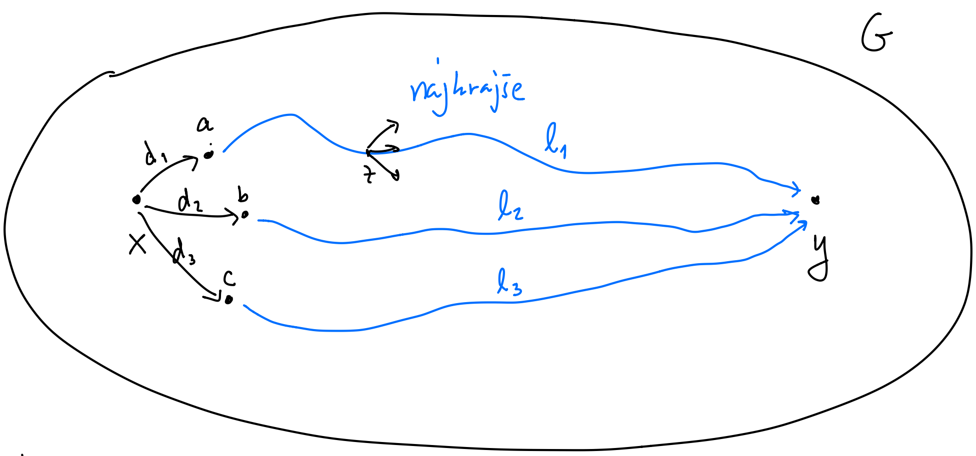
Najkrajše poti v grafu

G graf

poišči najkrajšo pot med danima vozliščema



$P(z)$ = najkrajša pot od z do y



$P(z)$ na podprobleme:
za vsakega soseda w vozlišča z obravnavamo $P(w)$

Slovar sosedov

```
g = { 'a' : [('b', 10), ('c', 20), ('f', 10)],
      'b' : [('c', 5), ('d', 10)],
      'c' : [('e', 7)],
      'd' : [],
      'e' : [('d', 4)],
      'f' : []
    }
```