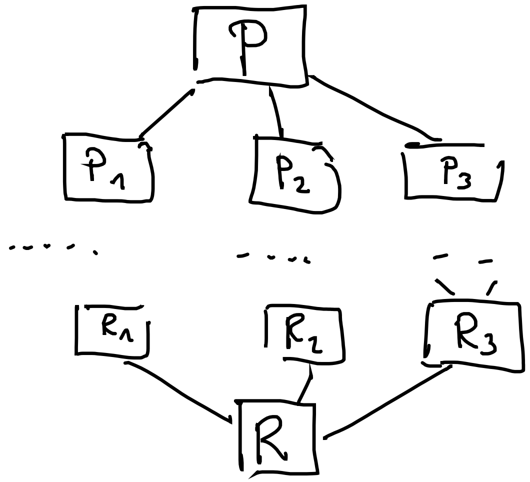
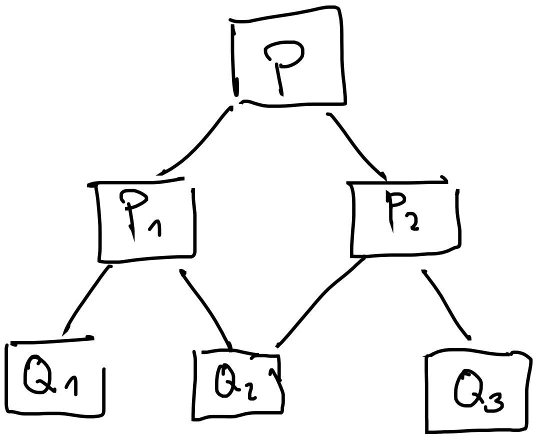
Dinamično programiranje

Spomnimo se: deli & vladaj

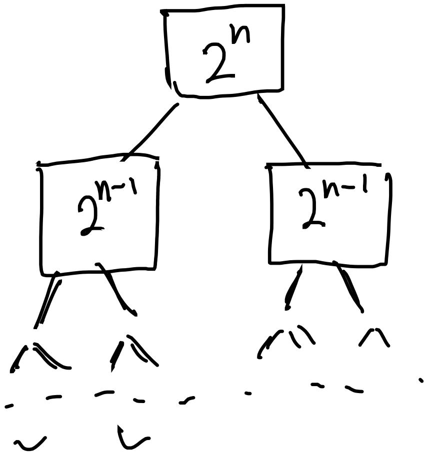




Potenca števila 2

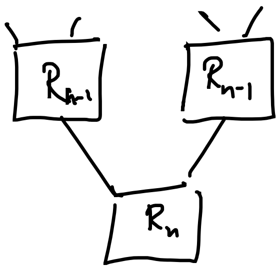
Naloga $P(n)$ = izračunaj 2^n

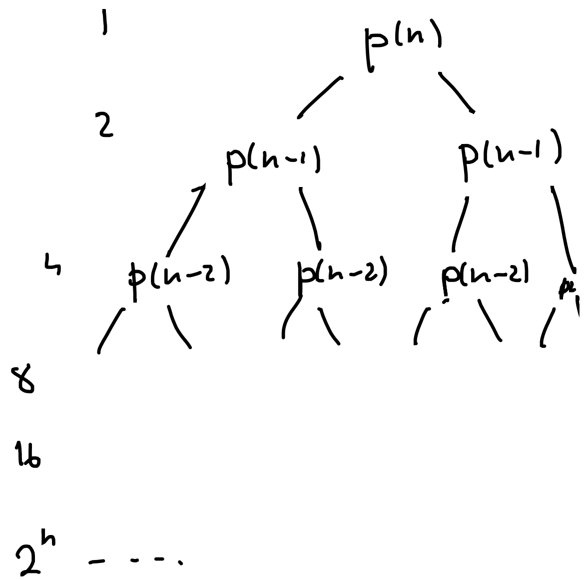
$$2^n = 2^{n-1} + 2^{n-1}$$




```
def p(n):
    if n==0: return 1
    else:
        a = p(n-1)
        b = p(n-1)
        return a+b
```

zahtevnost $O(2^n)$





Potenca števila 2

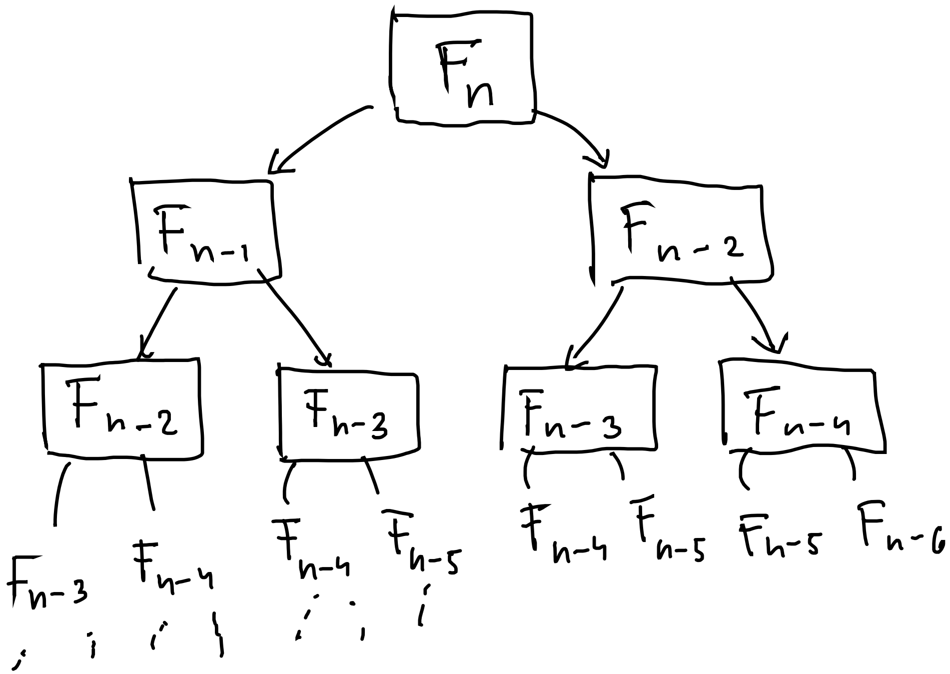

```
def p(n):
    if n==0:  return 1
    else:
        a = p(n-1)
        return a+a
```

zahtevnost $O(n)$

$p(n)$
$\downarrow$
$p(n-1)$
$\downarrow$
$p(n-2)$
$\vdots$
$\downarrow$
$p(0)$

Fibonacci

$F_0 = 0, \quad F_1 = 1, \quad F_{n+2} = F_{n+1} + F_n$



F_n 1

F_{n-1} 1

F_{n-2} ... 2

F_{n-3} ... 3

F_{n-4} 5

F_{n-5} ... 8

Memoizacija

Ko računamo P , si vedno zapomnimo vrednosti, ki smo jih že izračunali. Vedno najprej pogledamo, ali je vrednost, ki jo potrebujemo, že shranjena.

Primer: def $f(x)$:

$f(x_1)$
$f(x_2)$ } rekurzivni klic

Kako spravimo rezultate?

$f(x)$ .... v tabelo dodamo $[\underbrace{\ldots\ldots}_{\text{že znani rezultati}}, (3,2)]$

$fib(3) = 2$

Asociativni seznam: $[(\underset{\text{ključ}}{x_1}, \underset{\text{vrednost}}{y_1}), (x_2, y_2), \ldots, (x_n, y_n)]$

<u>Slovar</u>: $\{x_1 : y_1,$
$x_2 : y_2,$
$\vdots$
$\}$

Fibonacci s tabelo

F_n

0	1	2					n
F_0	F_1	F_2	F_3	…	F_{n-2}	F_{n-1}	F_n

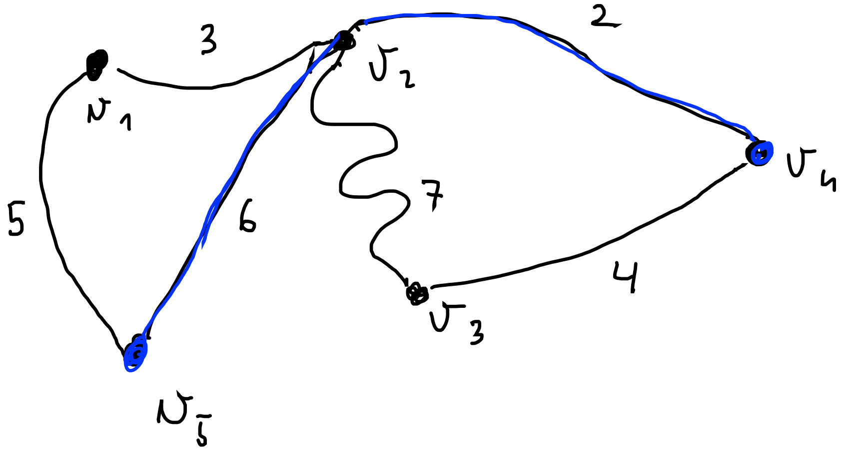
0	1	1	2	3	5	8	13	

Binomski koeficienti: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ $\qquad \binom{n}{0} = 1 \quad \binom{n}{n} = 1$

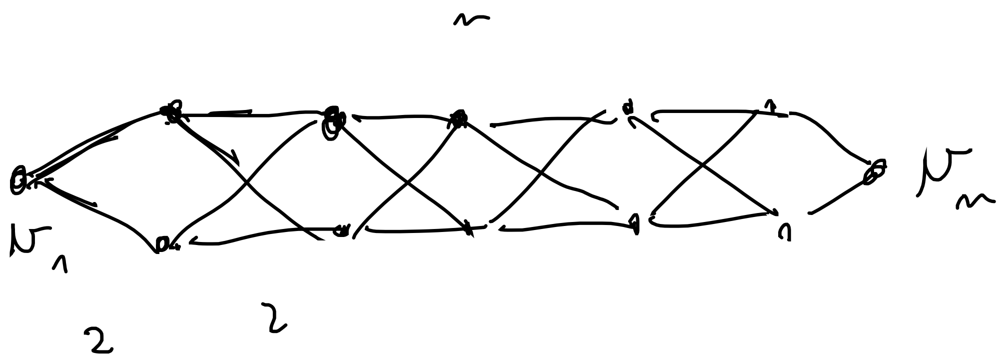
$k > n: \binom{n}{k} = 0$

k \ n	0	1	2	3			
0	1	1	1	1	1		
1	0	1	2	3	4		
2	0	0	1	3	6		
3	0	0	0	1	4		

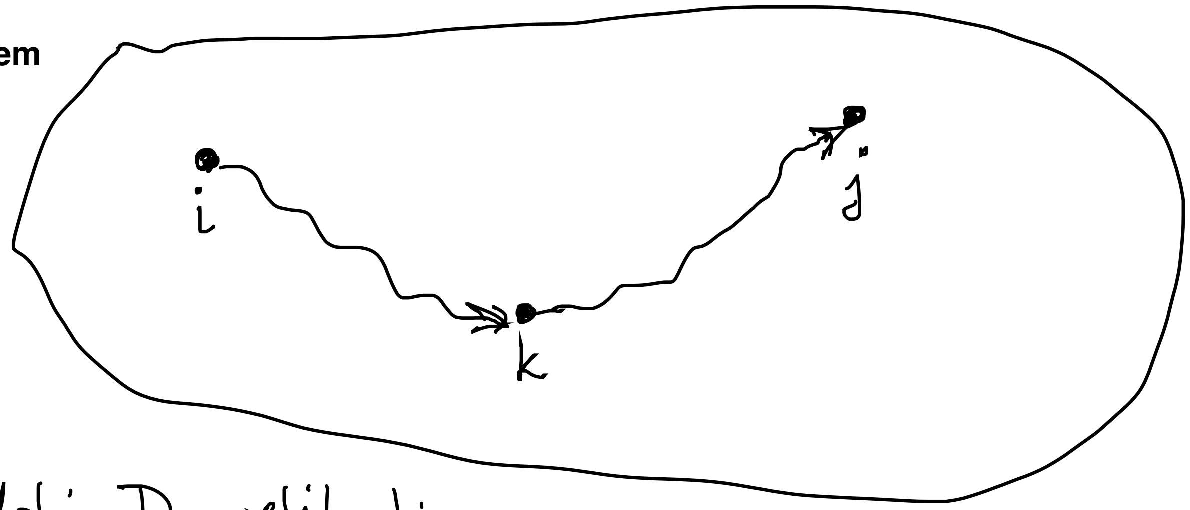
Najkrajše poti v grafu



$v_5 \longrightarrow v_4$?



Floyd-Warshallov algoritem



graf:
vozlišča
$0, 1, \ldots, n-1$

Matrika razdalj D velikosti $n \times n$

$D_{i,j}$ = razdalja med vozliščema i in j

$D_{i,i} = 0$

$D_{i,j} = D_{j,i}$ simetrične razdalje

$D_{i,j} = \infty$ če med i in j ni povezave

Problem VHOD: Matrika razdalj D v grafu z vozlišči $0, 1, \ldots, n-1$

IZHOD: Matrika dolžin najkrajših poti P

$P_{i,j}$ = dolžina najkrajše poti od i do j

<u>Ideja</u>: Matriko P računamo postopoma.

→ Najprej vanjo shranimo D (prekopiramo)

→ Potem računamo najkrajše poti od i do j, ki gredo skozi k, kjer teče k od 0 do n.

$Q_{i,j,k}$:= dolžina najkrajše poti med i in j, ki poteka skozi nekatera vozlišča $0, 1, 2, \ldots, k-1$

$P_{i,j} := Q_{i,j,n}$

$$Q_{i,j,0} := D_{i,j}$$

$$Q_{i,j,k+1} := \min\left(Q_{i,j,k},\ Q_{i,k,k} + Q_{k,j,k}\right)$$