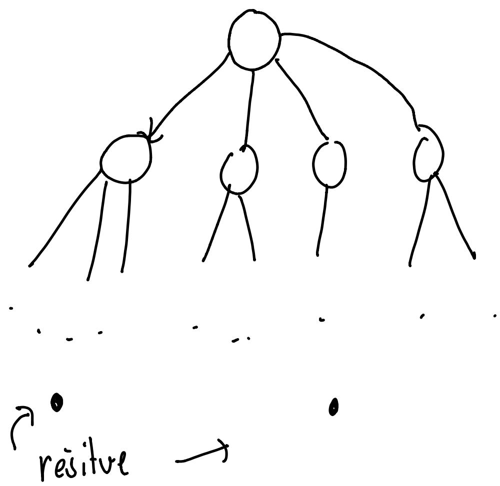


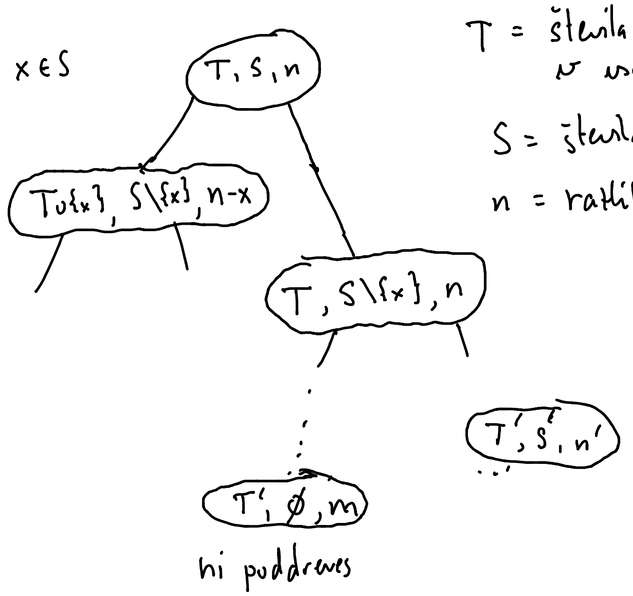
iščemo v:

- globino: najprej pregledamo poddrevesa
- širino: po nivojih

Primer: S množica celih števil, $n \in \mathbb{Z}$

Poišči $T \subseteq S$, da je $\sum T = n$.

$x \in S$



rešitev $\Leftrightarrow n' = 0$

T = števila, ki jih uporabimo v vsoti

S = števila, ki so na voljo

n = razlika do ciljne vsote

Če ima S k elementov:

- drevo ima $k+1$ nivojev
- pregledati moramo vozlišč:

$$1+2+4+\dots+2^k = 2^{k+1}-1$$

Računska zahtevnost

Koliko časa, prostora, komunikacije, energije, naključje ("resources") potrebuje program?
virov

Zanima nas ocena (velikostni red, asimptotično obnašanje), kako se neki algoritem/postopek obnaša:

- Koliko časa potrebuje v najslabšem primeru?
  povprečnem
  na realističnih podatkih

Časovna zahtevnost

merimo jo v "računskih korakih"

1 računski korak < konstanta C
↳ ocenimo na konkretnem računalniku

"Algoritem potrebuje $3n^2 + 5n$ korakov na podatkih velikosti n."

Dejanski čas $\leq C \cdot (3n^2 + 5n)$

Zapis "veliki O"

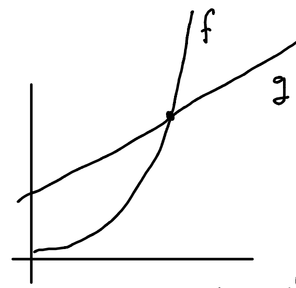
$f : \mathbb{N} \to \mathbb{R}_{\geq 0}$

$O(f) := \{ g : \mathbb{N} \to \mathbb{R}_{\geq 0} \mid \exists c > 0 . \exists n_0 \in \mathbb{N} . \forall n \geq n_0 . \, g(n) \leq c \cdot f(n) \}$

$g \in O(f)$ "g ne preseže $c \cdot f$ za neko konstanto c (za dovolj velike vrednosti argumenta)"

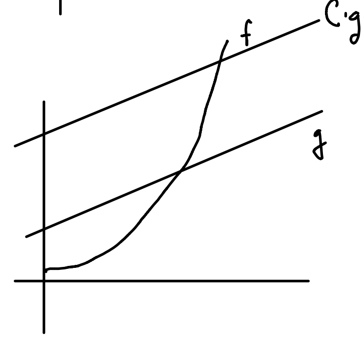
Primer:

$f(n) = n^2$ $\qquad g(n) = 20 \cdot n + 17$



$g \in O(f)$ ✓ od nekje naprej je $g(n) \leq f(n)$

$f \in O(g)$? ✗



Primer: $f(n) = 3n^2 + 50n$

$g(n) = n^2$

$g \in O(f)$ ✓

$f \in O(g)$? ✓ ker $3n^2 + 50n \leq \underset{50n^2 + 50n^2}{\underset{\shortparallel}{100n^2}}$ za $n \geq 1$

Primer: $f \in O(1)$ $\quad f(n) \leq C$ za dovolj velike n

Konstantni čas $\quad \Leftrightarrow f(n) \leq C'$ za vse n

n = velikost vhodnih podatkov

$$O(1) \subseteq O(\log n) \subseteq O(n) \subseteq O(n \cdot \log n) \subseteq O(n^2) \subseteq O(n^3) \subseteq \dots \subseteq O(2^n) \subseteq \dots$$

- $O(1)$: konstantni čas
- $O(\log n)$: zelo zelo učinkovito
- $O(n)$: zelo učinkovit
- $O(n \cdot \log n)$: učinkovito
- $O(n^2)$: meh!
- $O(n^3)$: ni uporabno
- $O(2^n)$: iluzorno

$n^2 + 1000n$ $\quad n = 10^4$: $\quad 10^8 + 10^7$

$\quad 10^5 \quad 10^{10} + 10^8$

Bisekcija

Naloga: vhod: urejena tabela števil in x

$$a = [a_0, a_1, a_2, \dots, a_{n-1}]$$

$$a_0 < a_1 < a_2 < \dots < a_{n-1}$$

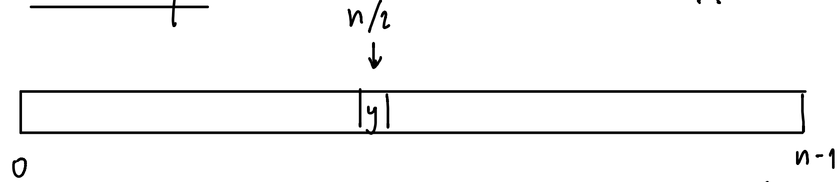
izhod: indeks k, da je $a_k = x$

"Poišči indeks elementa x v tabeli a"

Naivna rešitev: pregledamo tabelo po vrsti $\quad O(n)$

Bolje: <u>bisekcija</u>

x



nadaljuj iskanje tukaj, če je $x<y$ — $x=y$ ✓ — če je $y<x$

<u>z istim postopkom</u>

Iskalno območje:

0 .... n-1	širina n
....	$n/2$
....	$n/4$
	$n/8$
....	$n/2^k$
....	
...	1

Kolikrat moramo prepoloviti n, da dobimo 1?

$$\frac{n}{2^k}=1 \Rightarrow k=\log_2 n$$

$$O(\log_2 n)$$