$f: \mathbb{N} \to \mathbb{R}$

$f \in O(g) . \exists M > 0 . \exists N \in \mathbb{N} . \forall n > N . \; f(n) \leq M g(n)$

množenje $n^3 \wedge n \log_2 n^7 \wedge n$

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0 \Rightarrow f \in O(g)$$

$\forall \varepsilon > 0 . \exists N > 0 . \; x > N . \Rightarrow \left| \frac{f(x)}{g(x)} \right| < \varepsilon$

$\Rightarrow |f(x)| < \varepsilon |g(x)|$

potenje

$\exists \varepsilon > 0 . \exists N > 0 . \forall n > N . \; f(n) \leq \varepsilon g(n) \Rightarrow f \in O(g)$

$\| \; \uparrow$

$o(1) \subset O(\ln(\ln(n)))$

$O(\log_2(n)) = O(\log_{10}(n))$

$\log_2(n) = \frac{\log_{10} n}{\log_{10} 2} \in O(\log_{10}(n))$

$\log_{10}(n) = \frac{\log_2 n}{\log_2 10} \in O(\log_2(n))$

kolena

- že celo poljotje (predvsem po kolesu...)
- zdaj še po kolenu jemlje na fotoj
- sedaj ima obrabljen hrustanec

$$\sigma(1) \subset \sigma(\log(\log n)) \subset \sigma(\log_2(n)) = \sigma(\log_{10}(n))$$

$$\subset \sigma(\sqrt{n}) \subseteq \sigma(\log(n!)) \subseteq \sigma(n \log n) \subseteq \sigma(3^{\ln n})$$

$$\subseteq \sigma(n^2) \subseteq \sigma(2^n)$$

$$\subseteq (2^{2\log n})$$

$$\sigma(\log_a(n)) \subset \sigma(n^{\varepsilon})$$

$$\log(n!) \leq \log(n^n) = n \log n$$

$$\sigma(\log(\log(n))) \subseteq \sigma(\log n)$$

$$\log(n) < n$$

in log

inj neraščajoča

$$\log(\log(n)) \leq \log(n)$$

$$\log(n) \in \sigma(\sqrt{n})$$

$$\lim_{n \to \infty} \frac{\log n}{\sqrt{n}} \overset{LH}{=} \frac{\frac{1}{n}}{\frac{1}{2}n^{-\frac{1}{2}}} = \ln 2 \frac{1}{2\sqrt{n^3}} = 0 \Rightarrow \log(n) \in \sigma(\sqrt{n})$$

$$3^{\ln(n)} = 3^{\frac{\log_3(n)}{\log_3 e}} = = n^{\frac{1}{\log_3 e}} = n^{\varepsilon > 1}$$

$$\sqrt{n} < n < n \log n$$

$$O(n \log n) = O(\log n!)$$

$$\log n! = \sum_{i=1}^{n} \log i$$

$$\log n! > \log \left(\frac{n}{2}\right)^{\frac{n}{2}} = \frac{n}{2}(\log n - \log 2) \geq n($$

$$n \log n < n^2$$

$$n \log n \leq n \cdot n \leq n^2$$

$$O(n^2) \subseteq O(2^n) \subseteq O(2^{n \log n}) \subseteq O(n!)$$

$$2^{\log n^n} = 2^{\frac{\log_2 n^n}{\log_2 10}} = n^{\log_2 10 \, n}$$

seznam: n

append: $O(1)^*$

delete(i) $O(n)$ $(n-i)$

copy: $O(n)$

upddate(i): $O(1)$

get(i): $O(1)$

find(a): $O(n)$

add: $O(n)$

le nes tipična zanim.	
add(i)	$O(n)$
add(0,n)	$O(n)O(1)$
delete(;)	$O(n)$
delet(a,n)	$O(n)O(1)$
get(;)	$O(1)$
search(x)	$O(n)$

le nes tipična zanim.	časa	prostor
add(i)	$O(n)$	$O(1)$
add(0,n)	$O(n)O(1)$	$O(1)$
delete(;)	$O(n)$	$O(1)$
delet(a,n)	$O(n)O(1)$	$O(1)$
get(;)	$O(1)$	$O(1)$
search(x)	$O(n)$	$O(1)$

Slovar

$O(1)$ vse

algoritem za iskanje maximuma

```
m = seznam[0]
for c in seznam[1:]
    if c > m ⇒ m = c
return m
```

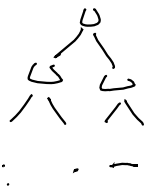
$O(n), O(1)$

```
seznam.sort
seznam[-1]
```

$O(n \log n)$ $O(1)$

$$\sum_{i \in N}^{n} i = O(n^2)$$

Rekurzivno: seznam na dva dela



$\log n$ globina

čas je lahko $O((\log n)$ z n procesorji

lahko se ji tudi $O(\frac{n}{\log n})$ pravilno injo

še vedno hitreje

Bitne operacije

- bitand &
- bitor |
- bitxor ^
- bitnot !, ~
- bitshiftleft <<
- bitshift right >>

V jeziku
&&
||
!=
not

z: 1 1 1
g: 1 0 0 1

$$\begin{array}{r} 1\,1\,1 \\ 1\,0\,0\,1 \\ \hline 1 \end{array} \;\&$$

$$\begin{array}{r} 0\,1\,1\,1 \\ 1\,0\,0\,1 \\ \hline 1\,1\,1\,1 \end{array} \;|$$

$$\begin{array}{r} 1\,1\,1 \\ 1\,0\,0\,1 \\ \hline 1\,1\,1\,0 \end{array} \;\hat{}$$

$$!\;\begin{array}{r} 1\,1\,1 \\ \hline 0 \end{array}$$

$$!\;\begin{array}{r} 1\,0\,0\,1 \\ \hline 1\,1\,0 \end{array}$$

$$<<\begin{array}{r} 1\,1\,1 \\ \hline 1\,1\,1\,0 \end{array}$$

$$<<\begin{array}{r} 1\,0\,0\,1 \\ \hline 1\,0\,0\,1\,0 \end{array}$$

$$g >> 2\;\begin{array}{r} 1\,0\,0\,1 \\ \hline 1\,0 \end{array}$$

$a \oplus b \oplus b = a$

a \ b	0	1
0	$0\oplus0=0$	$1\oplus1=0$
1	$1\oplus0=1$	$0\oplus1=1$

a...bit

$$\begin{array}{r} a_n \ldots a_2 a_1 a_0 \\ \& \quad [0 \ldots 01\ 0 \ldots 0] \leftarrow \text{mask} \\ \hline 0 \ldots a_i \ldots 0 \end{array}$$

if (a & mask)

> 0 zeto True

mask = 1 << i

1.) [0 . . . . . 0]

~ [1 . . . . . 1]

>> n-i [0 . . . . 1 . . 1]

~((~0) << i)

(1 << i) - 1

funkcija spejne a, i

$a_n \ldots a_1 a_0 \rightarrow a_n \ldots 1 \ldots a_1 a_0$

```
def f(a,i):
    mask := 1<<i
    return a | mask
```

```
def f'(a,i)
    mask := ~(1<<i)
    return a & mask
```

n = 0 0 1 0 1 0 1 1 0 0 0 0 1 1 0 1 0 1 0

10 mesto

levo: n-1
desno: n+1
gor: n-4
dol: n+4

```
def levopogj(a)
    maskp = (1<<4)-1
    p = a & maskp
    mask = ~(1<<20-p10+1)
    return (a & mask) - 1
```

ali a = a & mask
a = (a>>4)<<4
return a | p

```
def fib(n):
    prv = 1
    drv = 1
    for i in range(n)
        drv = drv + prv
        prv = drv - prv
    return drv
```

$O(drv) = 1{,}6^i = 1{,}6^n$

↑ velikost

št bitov za $drv = \log_2 1{,}6^i$

$= O(i)$

assign $= O(n)$

$$\sum v_1 + v_2 + v_3 = \sum_{i=1}^{n} 3O(i) = O(n^2)$$

TRIE – prefix tree