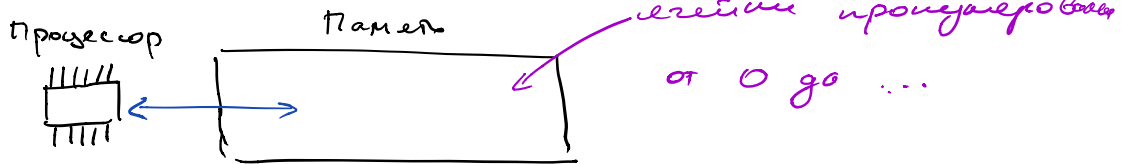


Алгоритм

↳ Модель вычисления

1. Машина Тьюринга
2. λ -исчисление
3. Алгоритмы Маркова
4. Машина Минского
- ⋮

Word RAM - машина



Доступ к \forall элементу памяти за 1 такт (1 операция)

w - # битов в одной ячейке

M элементов памяти $w \geq \log M$

Характерное значение $w \geq c \cdot \log n$

Базовые операции:

1. Ограничение к памяти (по номеру ячейки)
2. Арифметические операции с целыми числами порядка до 2^w (битовые операции тоже)
3. Арифметика с веществен. числами
4. Условные операторы и т.п.

YTB $F_n \geq 2^{n/2}, n \geq 6$

▷ База: $F_7 = 13 \geq 2^{3.5}$ $F_6 = 8 \geq 2^3$

$2^{3.5} = 2^3 \cdot 2^{0.5} = 8 \cdot \sqrt{2} < 8 \cdot 1.5 = 12$

Предположение: $\forall n \leq k, F_n \geq 2^{n/2}$

Переход: $F_k = F_{k-1} + F_{k-2} \geq 2^{(k-1)/2} + 2^{(k-2)/2}$
 $\geq 2 \cdot 2^{(k-2)/2} = 2^{(k-2)/2 + 1} = 2^{k/2}$ Δ

$2^n \geq F_n \geq 2^{n/2}$

⇒ Время работы fibs равен экспоненциально

$F_n \approx 1.6^n$

$F_n \approx \varphi^n$

$\varphi = \frac{\sqrt{5} + 1}{2}$

def fib2(n):

if n <= 2:

return n

количество операций

a[n]

$\leq 10n + c$

a[0] = 0

время работы:

a[1] = 1

$\leq n + c$

for i = 2 to n:

$a[i] = a[i-1] + a[i-2]$

return a[n]

def fib3(n):

if n <= 2:

return n

$$a = 0$$

$$b = 1$$

for $i = 2$ to n :

$$t = a + b$$

$$a = b$$

$$b = t$$

return b

операций:

$$\leq 10n + c$$

стек памяти:

$$\leq 6$$

«Сложность алгоритма fib2 - линейная»

O -символика

$$\exists f, g: \mathbb{N} \rightarrow \mathbb{R}_{>0}$$

• $f(n) = O(g(n))$ « f растёт не быстрее g »
 $f \leq g$

$$\exists c > 0: \forall n \ f(n) \leq c \cdot g(n)$$

Сложность fib2 - $O(n)$

$$\bullet \quad 30n + 155 = O(n)$$

$$c = 200$$

$$\rightarrow 30n + 155 \leq 200n$$

$$\bullet \quad 13n^2 + 24n + 105 = O(n^2)$$

$$\underline{13n^2} + \underline{24n} + \underline{105} \leq 142n^2 = \underline{13n^2} + \underline{24n^2} + \underline{105n^2}$$

Общие правила:

- Можно выбрасывать нульчлики и адд. конст.
- Если полином, то важен только старший член.
- $n^a = O(n^b)$, $\forall a \leq b$
- $\log^c n = O(n^a)$, $\forall a, c \geq 0$ $\log^{1000} n = O(n^{0.001})$
- $n^a = O(c^n)$, $\forall a \geq 0, c > 1$
- $O(1)$

$\forall f, g: \mathbb{N} \rightarrow \mathbb{R}_{>0}$

$f(n) = \Omega(g(n))$ " f растёт не медленнее g "
 $\exists c > 0 : \forall n \quad f(n) \geq c \cdot g(n) \quad f \geq g$

Время работы $\text{fib}(n)$:
 $O(2^n) \quad \Omega(2^{n/2})$

Замечание:

$$f(n) = O(g(n))$$

~~$$\begin{array}{l} 2n = O(n) \\ 3n = O(n) \end{array} \quad \Bigg| \quad \Rightarrow \quad 2n = 3n$$~~

$$f(n) \in O(g(n))$$

Основные ресурсы: время и память