2

Сложност на алгоритъм

$$f(n) = 2n^2 + 300$$

големина на входа

брой стъпки, които алг. ще извърши.

Alg 1 vs Alg 2

$f(n)$ $\qquad$ $g(n)$

f vs g

Коя ф-я нараства по-бавно?

Интересуваме се при $n \to \infty$

При малко n, повечето алг. ще се справят добре!

$$\lim_{n \to \infty} \frac{f}{g}$$

∞ $\quad f > g$ $\quad \boxed{g}$

const $\quad f \approx g$ $\quad \boxed{f \mid g}$

0 $\quad f < g$ $\quad \boxed{f}$

Ако имаме ограничение на входа?

Пример: сортиране на масив, който има ≤ 250 ел.

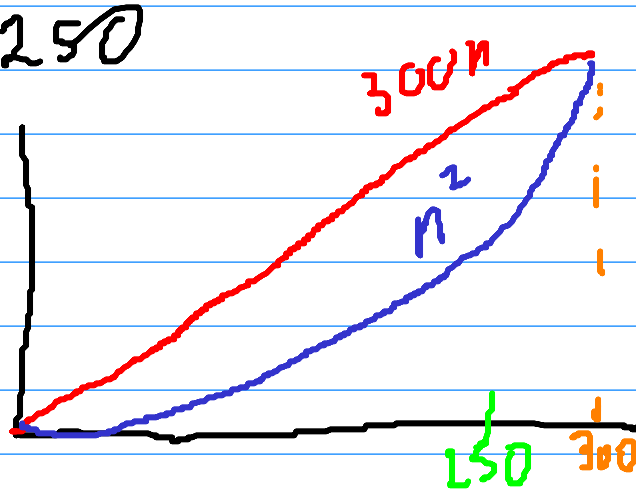
$n \le 250$

Alg 1	Alg 2
$\boxed{300 \cdot n}$	$\boxed{n^2}$

• При $n \to \infty$ $\boxed{\text{Alg 1 ✓}}$

$$\underbrace{300 \cdot n}_{\Theta(n)} < \underbrace{n^2}_{\Theta(n^2)}$$

• При $n \le 250$



$\boxed{\text{Alg 2 ✓}}$

Анализ на итеративни Алгоритми

- Чрез сумиране.

Общи правила:

$$\sum_{i=1}^{n} i^{a} \qquad \begin{array}{ll} a \geq 1 & \Theta(n^{a+1}) \\ a = -1 & \Theta(\log(n)) \\ 0 \leq a < 1 & \Theta(n) \end{array}$$

$$\sum_{i=1}^{n} i^{1} = 1 + 2 + 3 \dots + n = \frac{n(n+1)}{2} \simeq \Theta(n^2)$$

$$\sum_{i=1}^{n} i^{2} = 1^2 + 2^2 + \dots n^2 = \frac{n(n+1)(2n+1)}{6} \simeq \Theta(n^3)$$

$$\sum i^{3} = 1^3 + 2^3 + \dots n^3 = \frac{n^2(n+1)^2}{4} = \Theta(n^4)$$

$$\sum_{i=1}^{n} i^{-1} = \underbrace{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} \dots \frac{1}{n}}_{H_n \text{ – Хармонично число}} \qquad \Theta(\log(n))$$

$$\sum_{i=1}^{n} 1 = \theta(\log(n))$$

$i *= 2$

$$\underbrace{\underset{i=1}{1} + \underset{i=2}{1} + \underset{i=4}{1} + \underset{i=8}{1} \ldots + 1}_{\log(n)}$$

- Най-добър сл.
- ✓ Среден сл.
- Най-лош сл. — Смята се по-лесно от средния сл.

Задача: Анализирайте алгоритмите в най-лошия случай

- Ел. операция — отнема конст. време

! Не зависи от големината на входа!

зад 1 $\quad \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{n} 1}_{n} + \sum_{k=0}^{\frac{n}{2}} 1 =$

$$= \sum_{i=1}^{n} n + \frac{n}{2} =$$

$$n^2 + \frac{n}{2} \asymp \Theta(n^2)$$

зад 2

$i=1$	1
$i=2$	4
$i=3$	9
$\vdots$	

$$\overbrace{\sum_{i=1}^{n} i^2}^{\asymp\, \Theta(n^3)} + \Theta(1)$$

$$= \Theta(n^3)$$

зад 3

$$1+2+4+8 \dots + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n$$

$$= \sum_{\substack{i=n \\ i/=2}}^{1} \frac{n}{i} \asymp 2n-1 \quad \boxed{\Theta(n)}$$

$n = 8 \qquad 2n-1$

$1+2+4+8 = 15$

$n = 16 \qquad 2n-1$

$1+2+4+8+16 = 31$

3ag 4 $\downarrow$

3ag 5 $$\sum_{\substack{i=1 \\ i*=2}}^{n} \sum_{j=1}^{n} 1 = \sum_{\substack{i=1 \\ i*2}}^{n} n =$$

$$\underbrace{n + n + n \ldots \quad n}_{\log(n)} \qquad \Theta(n \cdot \log(n))$$

Ako $n=16$

$$\left.\begin{matrix} i=1 \\ i=2 \\ i=4 \\ i=8 \\ i=16 \end{matrix}\right\} \log(n)+1 = \Theta(\log(n))$$

~~3ag 6~~ $$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j+=i}}^{n} 1 = \sum_{i=1}^{n} \frac{n}{i} = n \underbrace{\sum_{i=1}^{n} \frac{1}{i}}_{\log(n)}$$

$$\Theta(n \cdot \log(n))$$

~~зад 7~~

$\underbrace{\Sigma\Sigma\Sigma\Sigma 1}_{\Theta(n^4)} + \underline{2^n} + \underbrace{const}_{\Theta(1)}$

$\boxed{\Theta(2^n)}$

! работа ull (64 бита)

2^{63} → горна граница

→ можем да приемем за константа

~~зад 4~~

за f: $\Theta(n)$

за task4 (n)

↓

извиква n пъти f

$\Rightarrow$ task4 → $\underbrace{f, f, f \dots f}_{n \text{ пъти}}$

$\cancel{\Theta(n^2)}$

През изпълнението на tasky променливата res е 0 или 1

Приемаме f в рамките на tasky за $\Theta(1)$

$$\sum_{\substack{i=n \\ i=1}}^{i=0} 1 = \underbrace{1+1+1\cdots+1}_{\log(n)} = \Theta(\log(n))$$

Стойности на i -> $\underbrace{n, \frac{n}{2}, \frac{n}{4} \cdots 1}_{\log(n)}$

Анализ на използвани алгоритми

1. Алгоритми за търсене

1) Линейно търсене

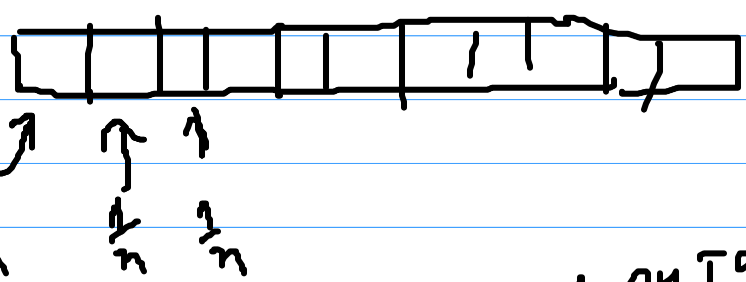
Най-добр. → безсмислена величина $\Theta(1)$

Средн.: ?? $\Theta(n)$

Най-лоша $\Theta(n)$

Търсим елементът x,
приемаме, че го има точно веднъж в масива

→ Къде може да е x ?



Вероятността елемента x да е на k-та позиция е $\frac{1}{n}$

Стъпки	1,	2,	3,	4 ...	n
вер.	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{n}$	$\frac{1}{n}$

Вероятността x да е на 3-та позиция

Пример за очаквана стойност

Хвърляне на зар -> брой точки?

Точки:	1	2	3	4	5	6
вер:	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

↳ очакваната стойност на хвърлените точки

$$1*\frac{1}{6} + 2*\frac{1}{6} + 3*\frac{1}{6} + 4*\frac{1}{6} + 5*\frac{1}{6} + 6*\frac{1}{6} = \boxed{3,5}$$

1000 зара $\quad \frac{\text{сума на точките}}{1000} \approx 3.5$

1 2 3 4 5 6

средно на хвърляне получаваме 3.5 точки

Връщаме се на пресмятане на ср. случай на линейното търсене

$$1*\frac{1}{n} + 2*\frac{1}{n} + 3*\frac{1}{n} \dots \quad n*\frac{1}{n} =$$

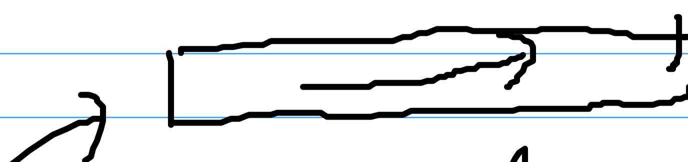
$$= 1 * \frac{1}{n} + 2 * \frac{1}{n} + \dots n * \frac{1}{n} =$$

$$= \frac{1 + \dots n}{n} = \frac{\frac{n(n+1)}{2}}{n} = \frac{n+1}{2} \approx \Theta(n)$$

среден брой стъпки

$\Rightarrow$ В средния случай: $\Theta(n)$

2/ Двоично търсене



сортиран масив.

Най-добре: $\Theta(1)$ елементът е в средата

Средно: Домашно $\Theta(\log(n))$

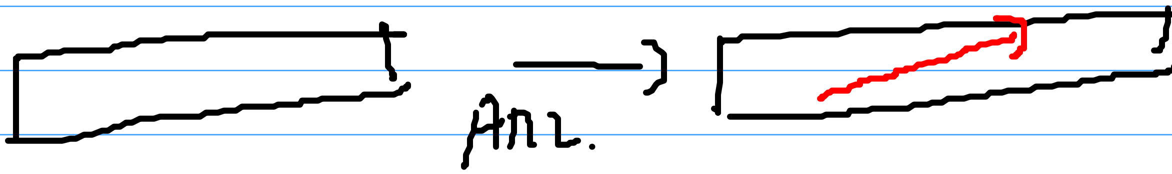
Най-лошо: $\Theta(\log(n))$

$$n \xrightarrow{\Theta(1)} \frac{n}{2} \xrightarrow{\Theta(1)} \frac{n}{4} \xrightarrow{\Theta(1)} \frac{n}{8} \xrightarrow{\Theta(1)} \dots \xrightarrow{\Theta(1)} 1$$

3) Търсене чрез скачане
4) експ. търсене
5) интерполационно търсене

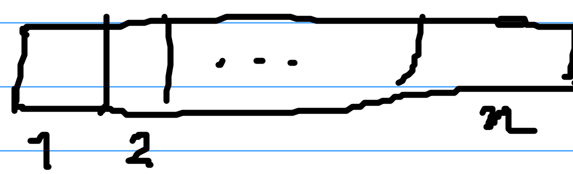
Не ги изучаваме

11. Анализ на Алгоритми за Сортиране



Алг.

arr[1... n]



def: Инверсия в масив

$i, j \in \{1 \dots n\}$

$i < j \wedge arr[i] > arr[j]$

1, 9, 6, 10

1 2 3 4

инверсия

$2 < 3 \wedge 9 > 6$

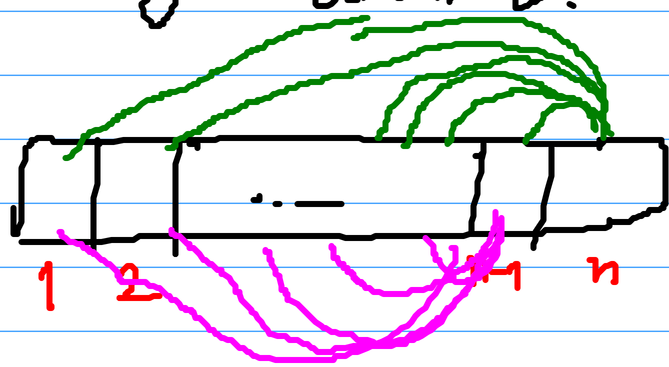
масив е сортиран, когато в него няма инверсии!

1, 9, 0, 5

Колко инверсии има?

3 инверсии

Колко най-много инверсии може да имаме?



Всички наредени двойки (i, j) ще са в инверсия

$$(n-1) + (n-2) + (n-3) \dots + 1 =$$

$$= \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

- Сортиране по метода на мехурчето (Bubble sort)

сложност време
- най-добро: $\Theta(n)$
- среден: $\Theta(n^2)$
- най-лош $\frac{n(n-1)}{2} = \Theta(n^2)$

- сложност по памет $\Theta(1)$

- стабилност

→ Ако масивът е сортиран наобратно:

=> имаме $\frac{n(n-1)}{2}$ инверсии

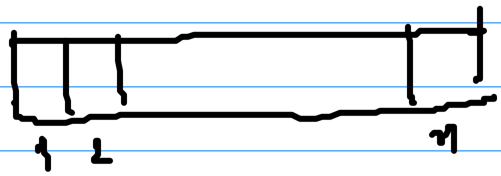
При една стъпка премахваме точно 1 инверсия

=> най-лош случай $\Theta(n^2)$

Анализ в средния случай:

Колко средно инверсии има в масив?

Нека Б.О.О. сортираме масив с числата {1 … n}

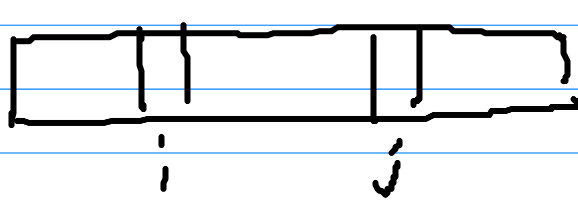


Възможните масиви са n!

Средно колко инверсии има в масива

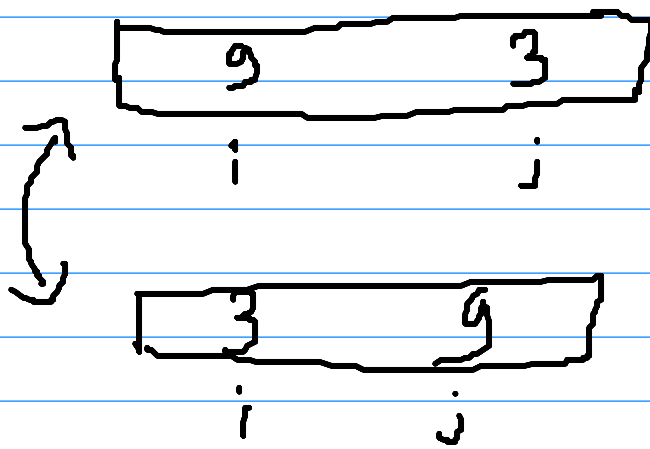
Произволна двойка от индекси (i, j)

i < j



Каква е вероятността (i, j) да е в инверсия?

$$\text{вероятност } (i,j) \text{ инверсия} = \frac{\text{перм. в които } (i,j) \text{ е инв.}}{\text{всички перм } (= n!)}$$



Има биекция му пермутациите, в които (i,j) е инверсия и пермутациите, в които (i,j) не е инверсия!

$\Rightarrow$ пермутациите, в които (i,j) е инверсия са точно половината $\left(\frac{n!}{2}\right)$

$\Rightarrow$ вероятността в произволен масив (i,j) да е инверсия $\frac{n!}{2} / n! = \frac{1}{2}$

Колко средно инверсии има в масив?

$\frac{n(n-1)}{2}$ · наредени двойка индекси (i,j) $i<j$
· възможности за инверсии

Всяка инверсия има вероятност $\frac{1}{2}$

$\Rightarrow$ Средно имаме $\underbrace{\frac{n(n-1)}{2}}_{\text{брой възможности за инверсия}} \cdot \underbrace{\frac{1}{2}}_{\text{вероятност за инв.}} =$

$= \frac{n \cdot (n-1)}{4}$

Среден брой инверсии.

Метода на мехурчето премахва на една стъпка една инверсия

стъпки $\geq \frac{n(n-1)}{4} = \Theta(n^2)$

$$\left| \underset{i<j}{\{(i,j)\}} \right| = \binom{n}{2}$$