

2

Сложност на алгоритъм

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

↑
коэффициент
на вжуга

↑
брой стъпки,
които алг.
ще извърши.

Alg 1 vs Alg 2
 $f(n)$ $g(n)$

$f \vee g$

Коя ф-я е по-добра и по-добра?

Интересувахме се при $n \rightarrow \infty$

при малко n , повечето алг.
повечето алг. ще се справят ок.

$$\lim_{n \rightarrow \infty} \frac{f}{g} \quad \begin{matrix} \infty & f > g & \boxed{g} \\ \text{const} + & f \sim g & \boxed{f, g} \\ 0 & f < g & \boxed{f} \end{matrix}$$

Ако имаме ограничение
на входа?

Пример: сортиране на
масив, който има ≤ 250 ел.
 $n \leq 250$

Alg 1

$$300 \cdot n$$

Alg 2

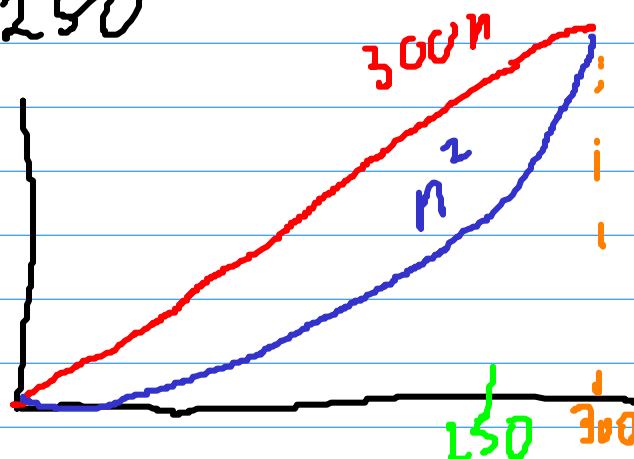
$$n^2$$

• при $n \rightarrow \infty$

Alg 1 ✓

$$\underbrace{300 \cdot n}_{\theta(n)} \prec \underbrace{n^2}_{\theta(n^2)}$$

• при $n \leq 250$



Alg 2 ✓

Анализ на итеративни алгоритми

- Чрез сумиране.
- Общи правила:

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

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

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

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

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

H_n - Хармонично
число

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

$i^* = 2$

$$\underbrace{1 + 1 + 1 + 1 + \dots + 1}_{\log(n)}$$

$i=1 \quad i=2 \quad i=4 \quad i=8 \quad \dots$

Най-голям сн.

✓ среден сн.

Най-малък сн — същото се
по-лесно

Задача: Анализирате Алгоритмите
в най-малкия случай

• Ен. операция — отнема конст. време

! Не зависи от големината
на входа!

log 1

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

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

$$n^2 + \frac{n}{2} \approx \theta(n^2)$$

log 2

$i=1$	1	$\sum_{i=1}^n i^2 \approx \theta(n^3)$
$i=2$	4	
$i=3$	9	
\vdots		

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

log 3

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

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

$$\begin{array}{l} n=8 \quad 2^{n-1} \\ 1+2+4+8=15 \end{array}$$

$$\begin{array}{l} n=16 \quad 2^{n-1} \\ 1+2+4+8+16=31 \end{array}$$

30y4



30y5

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n 1 = \sum_{\substack{i=1 \\ i \neq 2}}^n n =$$

$$\underbrace{n + n + n \dots n}_{\log(n)}$$

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

Ans

$$n=16$$

$$i=1$$

$$i=2$$

$$i=4$$

$$i=8$$

$$i=16$$

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

~~30y6~~

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n 1 = \sum_{i=1}^n \frac{n}{i} = n \sum_{i=1}^n \frac{1}{i}$$

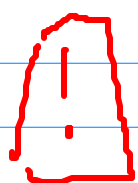
$$\log(n)$$

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

~~309 7~~

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

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



по сути 2^{128} (128 бит)

$2^{128} \rightarrow$ огромная Zahl

\rightarrow можно считать константой

~~309 4~~

за $f: \Theta(n)$

за task 4



избирая n раз f

$$\Rightarrow \text{task 4} \rightarrow \underbrace{f, f, f, \dots, f}_{n \text{ раз}}$$

~~$\Theta(n^4)$~~

През извършването на tasky
променливата res е 0 или 1

Приемаме f в рамките

на tasky $\theta(1)$

$$\sum_{\substack{i=0 \\ i=n \\ i=2}} 1 = \underbrace{1 + 1 + 1 \dots + 1}_{\log(n)} = \theta(\log(n))$$

Стойности
на i →

$$\underbrace{n, \frac{n}{2}, \frac{n}{4} \dots 1}_{\log(n)}$$

Анализ на използвания
Алгоритъм

①

Алгоритъм за търсене

1) линейно търсене

Най-добър \rightarrow безмислена величина $\theta(1)$

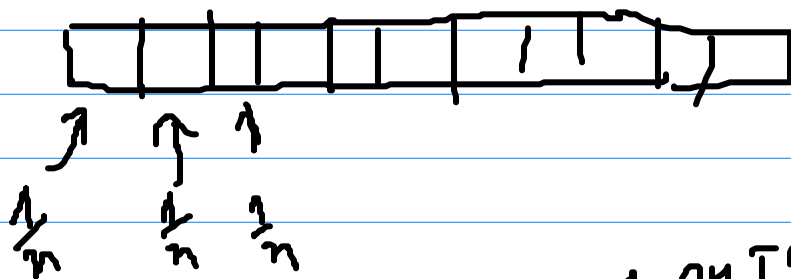
Средн. ?? $\theta(n)$

Най-лоша $\theta(n)$

Търсим елемент x ,

приемаме, че x има в масива.

\rightarrow къде може да е x ?



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

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

Вероятността
да е на 3-та позиция

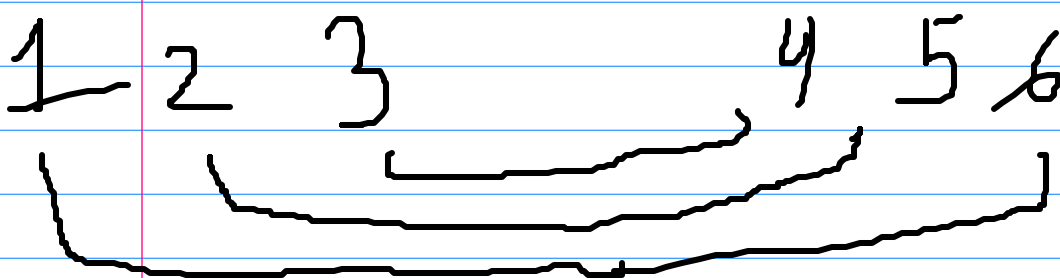
Пример за очакваната стойност
хвърляне на ЗНТ \rightarrow брое точки?

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

\hookrightarrow очакваната стойност
на хвърляните точки

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

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



средно
на хвърляне
на ЗНТ е
3.5 точки

Връщаме се на пресмятане на
бр. случаи на линеиното търсене

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

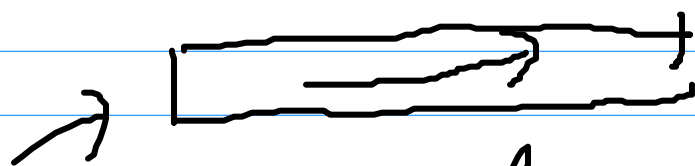
$$= 1 * \frac{1}{n} + 2 * \frac{1}{n} + \dots + n * \frac{1}{n} =$$

$$= \frac{1 + \dots + n}{n} = \frac{\frac{n(n+1)}{2}}{n} = \frac{n+1}{2} \sim \theta(n)$$

\Rightarrow В среднем
случае: $\theta(n)$

Среден
случ
случае

2/ Двоично търсене



сортиран масив.

наим-голям: $\theta(1)$ елементът е резултат

Среден: Домашка $\theta(\log n)$

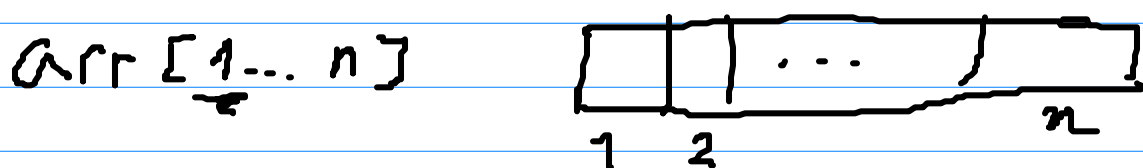
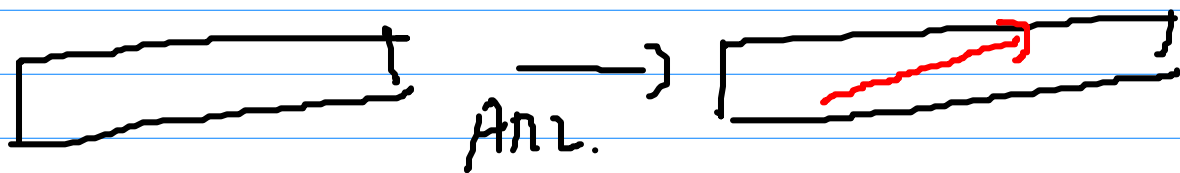
наим-малък: $\theta(\log n)$

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

$$\theta(1) \quad \theta(1) \quad \theta(1) \quad \theta(1) \quad \theta(1)$$

- 3) Търсене чрез скенер
 4) експ. търсене
 5) интерполяционно търсене
- } не се изучава

II Анализ на Алгоритми за сортиране



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

$$2 < 3 \wedge 9 > 6$$

маса е сортиран,
когато в него няма
инверсии!

1, 9, 0, 5

Колко инверсии има?

3 инверсии

Колко най-много инверсии
може да има?



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

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

сложност по място
 стабилност

$\Theta(1)$
 ⇒ Ако масивът е сортиран
 обратно:

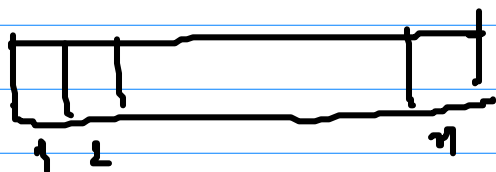
⇒ имаме $\frac{n(n-1)}{2}$ инверсии

при една стъпка
 правим точно 1 инверсия
 ⇒ най-лош случай $\Theta(n^2)$

Анализ в среднем случае:

Какая **Средно** инверсия има
в массив?

Нужно Б.О.О. сортировать массив
с числами $\{1 \dots n\}$



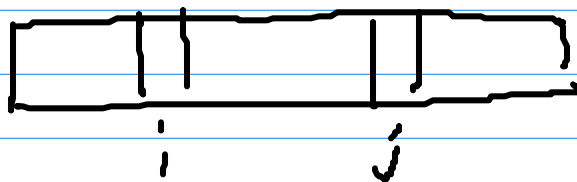
Возьмем такие массивы с n !

Средно кака инверсия
има в массива

Произведем выборку от n чисел

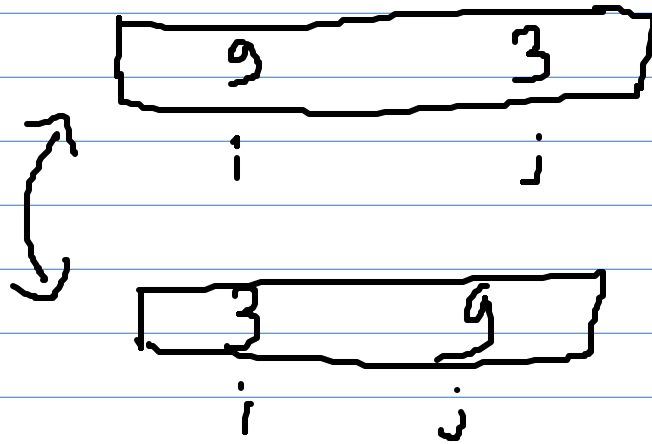
(i, j)

$i < j$



Кака 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 м/у 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 $(\frac{n!}{2})$

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

Колко средно инверсии има в масив?

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

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

\Rightarrow Средно имаме $\frac{n(n-1)}{2} \cdot \frac{1}{2} =$
Среден брой инверсии: $= \frac{n \cdot (n-1)}{4}$

брой възможности за инверсии
вероятност за инв.

Метод на мехурчето Премахва
на всеки стъпка едно инверсия

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

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