

Частичная сортировка

Задача: по массиву длины n вывести первые k позиций (самые большие) в упорядоченном виде.

1. $O(n \log n)$ — не использует $\forall O(n \log n)$ сортировку
Если данные "хорошие" \Rightarrow можно применить $O(n)$ сортировку.

2. Partial-Sort (A, k)

Make Heap (A) $O(n)$

for $i = 1$ to k

Extract Max (A) $O(\log n)$

$O(n + k \log n)$

Если $k = O(n / \log n) \Rightarrow O(n)$ алгоритм.

Порядковые статистики (nth element)

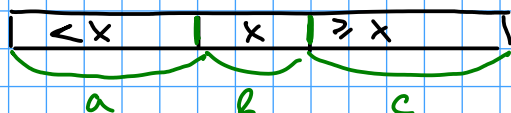
\equiv k -ая порядковая статистика массива A — это элемент, который будет занимать позицию с индексом k , если A упорядочить

\equiv Медиана — это $\lceil n/2 \rceil$ порядковая статистика

Randomized-Selection (A, i, j, k)

$x \leftarrow \text{Pivot}(A, i, j)$ // randomized

Partition(A, i, j, x)



if $k \leq a$:

Randomized-Selection ($A[<x]$, k)

else if $u \leq a+b$:

return x

else

Randomized-Selection ($A[1..n]$, $k-a-b$)

$$T(n) \leq T(n-1) + O(n) = O(n^2)$$

Сколько вызовов нужно в среднем, чтоб массив уменьшился в $4/3$ раза

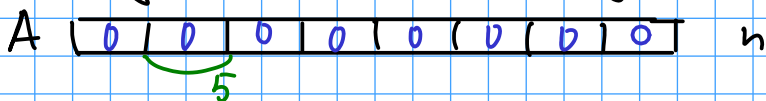


$$\begin{aligned} E(T(n)) &\leq E\left(T\left(\frac{3n}{4}\right)\right) + E(\# \text{вызовов} \cdot O(n)) = \\ &= E\left(T\left(\frac{3n}{4}\right)\right) + E(\# \text{вызовов}) \cdot O(n) = \\ &= O(n) \end{aligned}$$

Медiana медиан

Блок - Фрайг - Пратт - Ривест - Тарьян

1. Разобьем на подмассивы длины 5



2. Сортируем \neq подмассив

3. В \neq подмассиве выберем медиану \rightarrow

$n/5$ эл-ов A'

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|

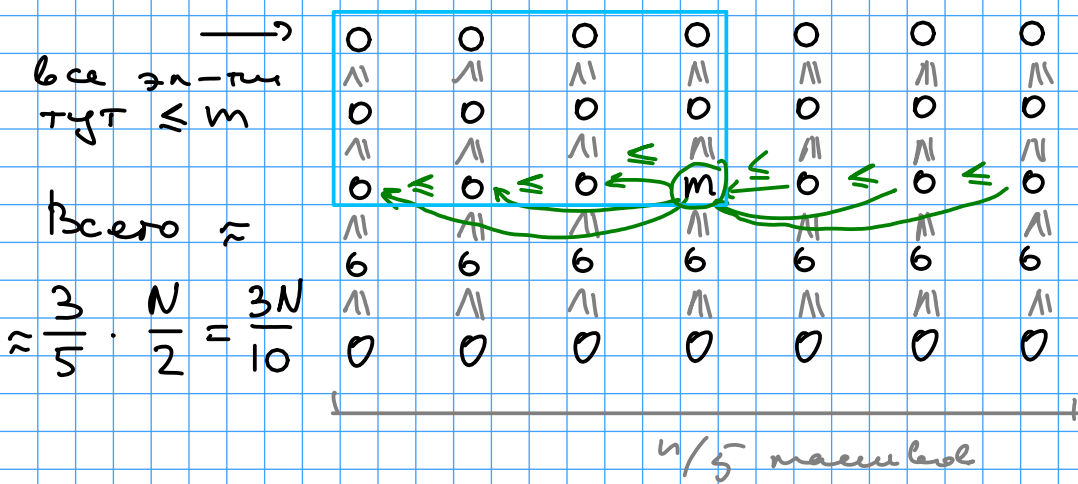
 $n/5$

4. Запускаем рекурсивно на $A' \rightarrow m$

5. Partition (A, m)

\nearrow
медиана
медиан

6. Рекурсивно вызываем на подмассиве.



Такая операция разделение в Partition (max 5)
 "отнимается" $\frac{3}{10}N \Rightarrow$ остаётся не более $\frac{7}{10}N$

$$T(N) \leq T\left(\frac{7}{10}N\right) + T\left(\frac{N}{5}\right) + O(N) \stackrel{?}{=} O(N)$$

$$\exists T(N) = O(N), \text{ т.е. } \exists c, d : T(N) \leq cN + d$$

$$T\left(\frac{7}{10}N\right) + T\left(\frac{N}{5}\right) + O(N) =$$

$$= c \frac{7}{10}N + d + c \frac{N}{5} + d + \alpha N + \beta =$$

$$= c \frac{9}{10}N + \alpha N + 2d + \beta \leq cN + d$$

$$\underbrace{\left(\frac{1}{10}c - \alpha\right)}_{>0} N \geq d + \beta$$

$c > 10\alpha \Rightarrow$ наймем c тем-ооо N
 это тем-во выполнимое

$c = 10\alpha + 10 \Rightarrow$ при $N \geq d + \beta$
 тем-во выполнимое.

Следствие:

Quick Sort с хорошей рекурсией работает
 \downarrow $O(n \log n)$ в худшем случае.

$$T(n) = 2T(n/2) + \Theta(n) \quad T(n) = \Theta(n \log n)$$

$$\exists T(n) = O(n) = cn + d$$

$$2T(n/2) + O(n) = cn + 2d + \alpha n + \beta \stackrel{?}{\leq} cn + d$$

$$-\alpha n \geq d + \beta \Rightarrow \text{нпротиворечие}$$