Alegerea medianei în O(n)

1. Algoritmul aleator
2. Algoritmul determinist

Problemă: De câte comparații avem nevoie pentru a găsi minimul și maximul dintr-un vector cu n elemente?

$n-1$

Dar dacă vrem să le găsim simultan?
Facem perechi de două elemente:

3   5   2 > 1   7 > 4   10

min = 3
max = 5

2 ? 1
1 ? min
2 ? max

min = 1
max = 7

min = 1
max = 5

$\frac{3}{2}n$ comparații
(3 comparații · $\frac{n}{2}$ perechi)

⇑

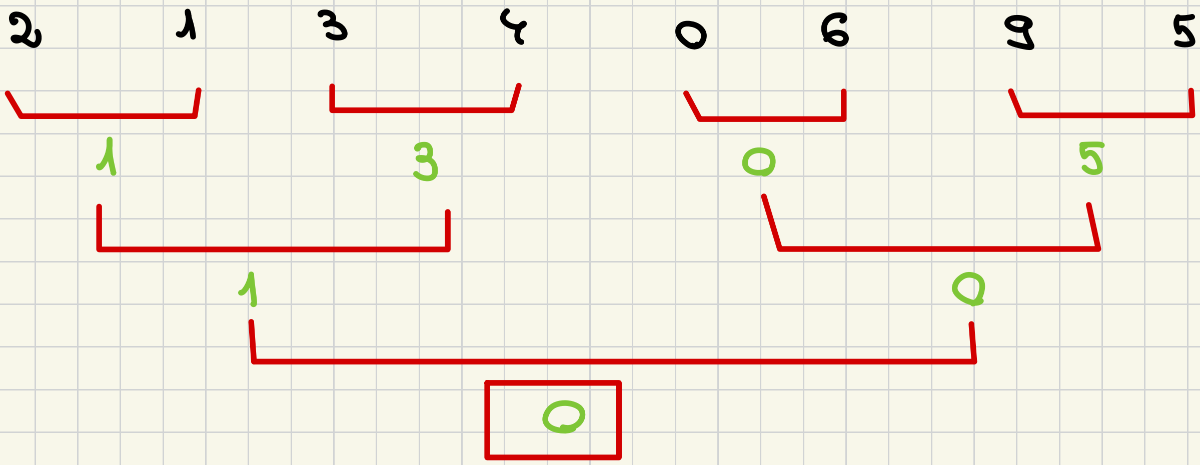
Comparăm cele 2 elemente între ele → 1 comparație
Comparăm mai micul (marele) cu min (max) → 2 comparații

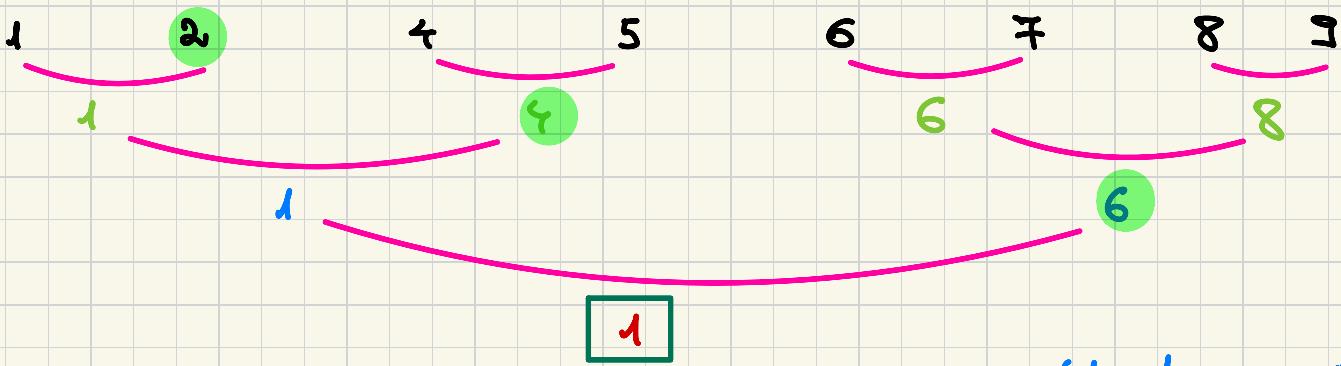
Problemă: De câte comparații avem nevoie pentru a găsi minimul și al doilea minim dintr-un vector?

1   3   2   4

min = 1
2 min = 3

$n + \log_2 n - 1$ comparații





$\frac{n}{2}$

$\frac{n}{4}$

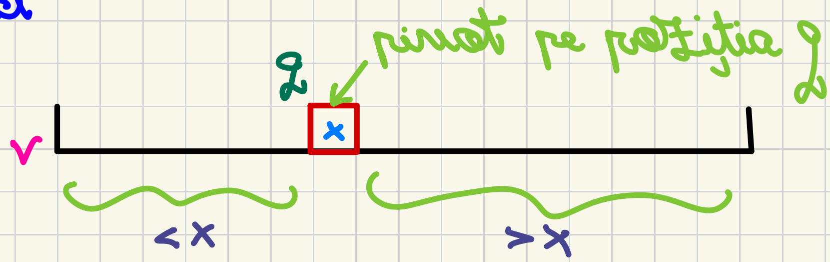
...

$n\left(\frac{1}{2}+\frac{1}{4}+\ldots+\frac{1}{2^k}\right)$

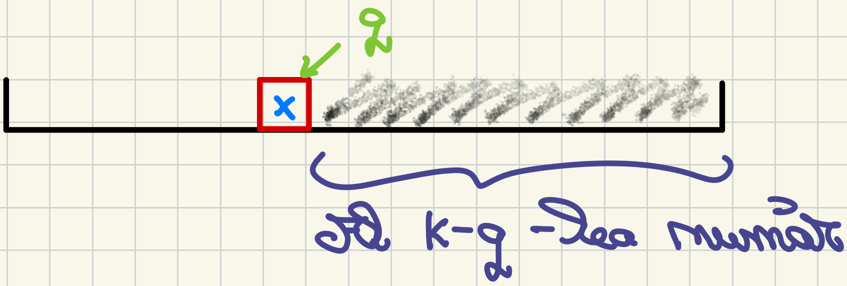
1. Algoritm aleator pentru găsirea medianei (în general, al k-lea element dintr-un vector)

Quick Select

1) $k < q$



2) $k > q$



pivot

1 7 3 $\boxed{5}$ 2 10 13 4 9 20

k=6

q=5

pivot

1 3 2 4 $\boxed{5}$ $\underbrace{7 \quad 10 \quad \boxed{13} \quad 9 \quad 20}_{k=1}$

7 10 9 $\boxed{13}$ 20

...se ia pivot până când q=k

q=4

vectorul

k-element (A, l, r, k)

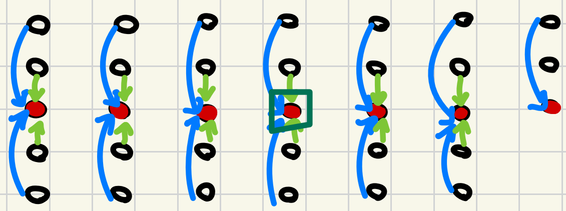
```
{
    q = Partiție (A, l, r)  → alege pivotul

    if (k == q)   return A[q]
    if (k < q)    return k-element (A, l, q-1, k)
    else return   k-element (A, q+1, r, k-q)
}
```

Caz favorabil: $T(n) = T(n/2) + O(n) = \Theta(n)$

defavorabil: $T(n) = T(n-1) + O(n) = \Theta(n^2)$

2. Algoritm determinist

1) Partiționăm numerele în grupe de câte 5



$\frac{n}{5}$ grupe

$\frac{n}{5} \cdot \frac{1}{2}$ au mediana mai mică decât mediana medianelor

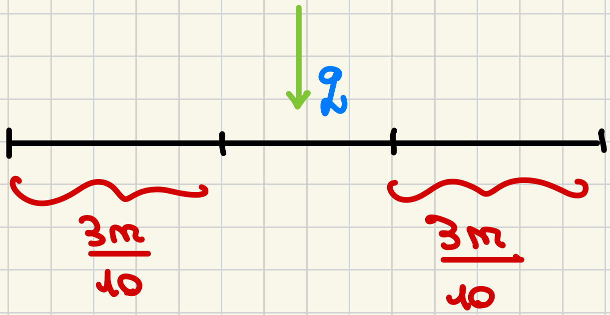
$\frac{3n}{10}$ elemente $\overset{\geq}{\leq}$ mediana medianelor

2) Găsim mediana din fiecare grupă (sortare)

3) Găsim mediana medianelor folosind același algoritm

4) Selectăm valoarea de la punctul 3 drept pivot

5) Continuăm în același mod ca algoritmul aleator



$$T(n) = \underbrace{T(n/5)}_{3} + \underbrace{T\left(\frac{7n}{10}\right)}_{5} + \underbrace{O(n)}_{1,2,4} = O(n) \quad \text{(pași)}$$

Arătăm că $T(n) \le c\,n$

Presupunem că $T(n/5) \le c \cdot \frac{n}{5}$

$$T\left(\frac{7n}{10}\right) \le c \cdot \frac{7n}{10}$$

$$T(n) = T(n/5) + T\left(\frac{7n}{10}\right) + n \le \frac{c \cdot n}{5} + c \cdot \frac{7n}{10} + n \le$$

$$\le c \cdot \frac{9n}{10} + n = c \cdot n - \frac{cn}{10} + n = cn - n\left(\frac{c}{10} - 1\right) \le cn,$$

$(\forall)\ c \ge 10$

Grupe de câte 3:

$$T(n) = T(n/3) + T\left(\frac{2n}{3}\right) + O(n) = \Theta(n \log n)$$

$\frac{n}{3}$ grupe

$\frac{n}{6}$ grupe $\times$ 2 elemente $= \frac{n}{3}$ elemente $\leq$ mediana medianelor

$$T(n) = T\left(\frac{n}{7}\right) + T\left(\frac{5n}{7}\right) + O(n) = \Theta(n)$$

Gruparea în 5 asigură echilibrul perfect între partiționare și cost de sortare.