

# Швидке сортування

Чолак Сергій

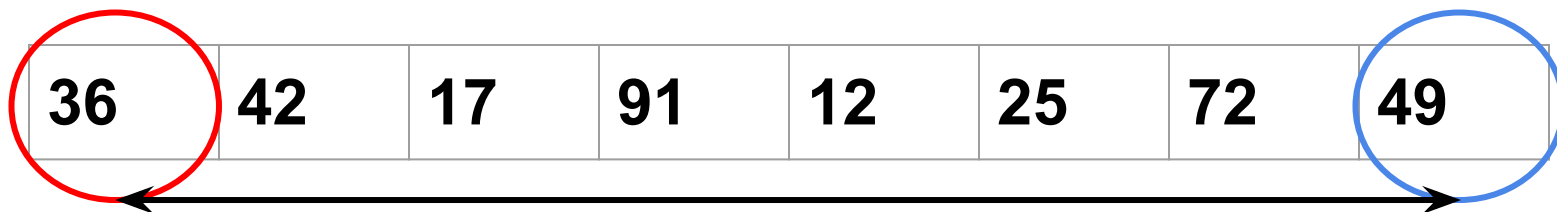
13.04.2018

# Історія та основи

- ↳ Тоні Гоар (Hoare) 1962
- ↳ Використовує в середньому  $O(n \cdot \log(n))$  операцій
- ↳ В найгіршому випадку робить  $O(n^2)$  порівнянь
- ↳ Швидше інших алгоритмів у своєму класі складності (групи обміну)

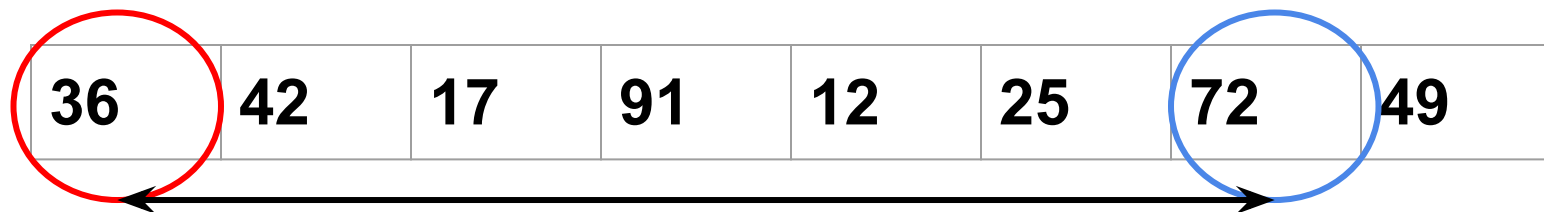
# Ідея

- Маємо масив. Обираємо медіану.
- Порівнюємо найвіддаленіші елементи



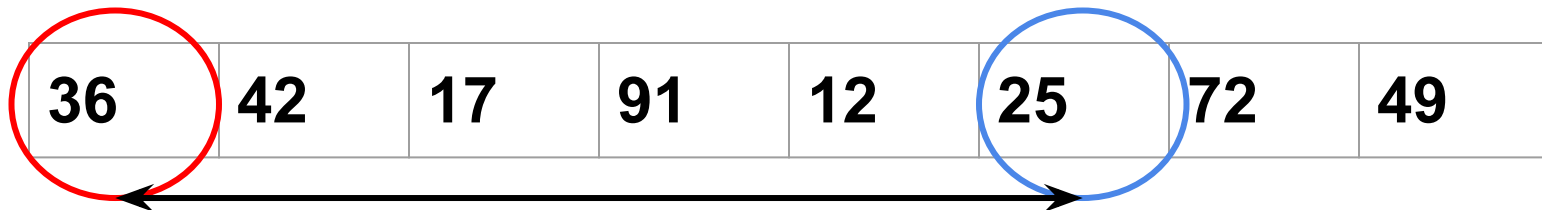
# Идея

→ Все ОК!



# Ідея

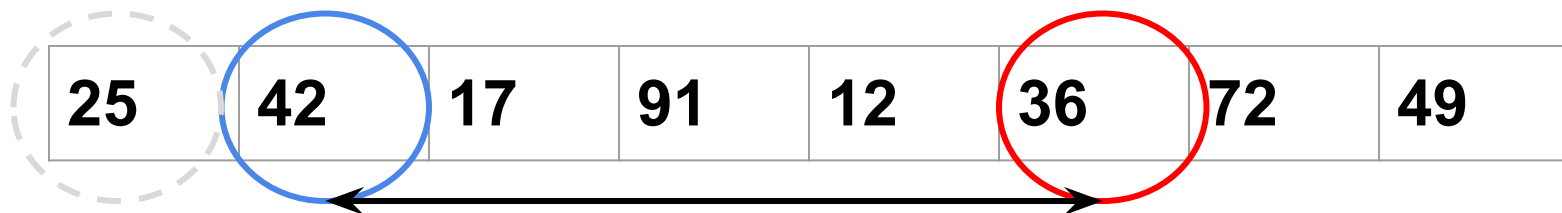
→ Міняємо місцями!



# Ідея

→ Порівнюємо з лівим кінцем

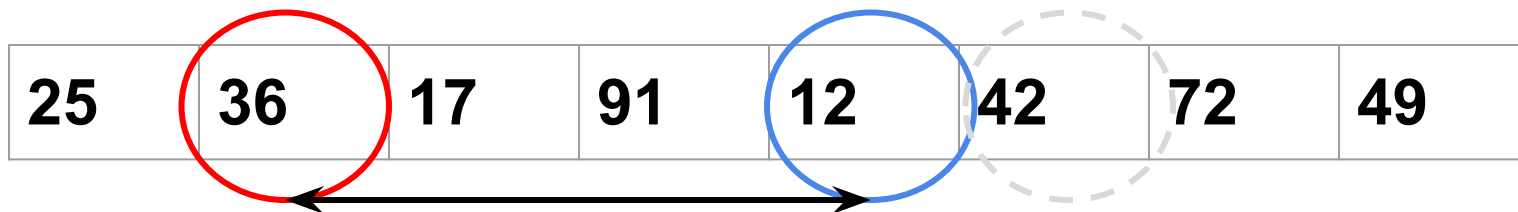
→ Міняємо!



# Ідея

→ Порівнюємо з правим кінцем

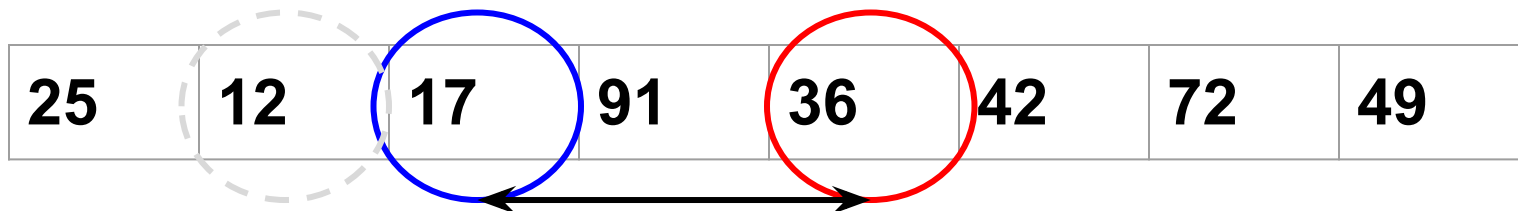
→ Міняємо!



# Ідея

→ Порівнюємо з лівим кінцем

→ Все ОК!

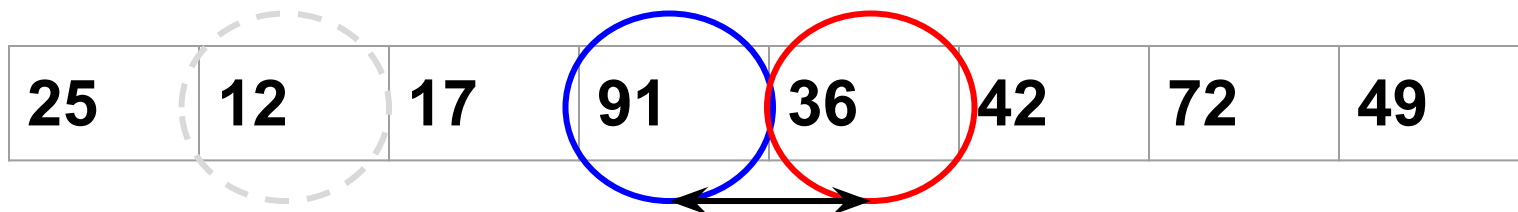




# Ідея

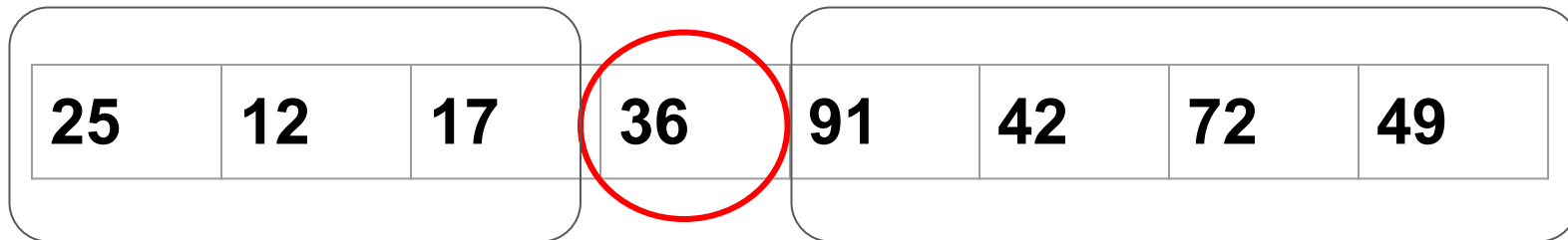
→ Порівнюємо з наступним

→ Міняємо!



# Ідея

→ Медіана на своєму місці. Більше не рухаємо її!



→ Зліва всі елементи менші, ніж **36**. Справа - більші

→ Повторюємо рекурсивно наші дії з половинками

# Реалізація

```
Quicksort( $A, p, q$ )
1 if  $p \geq q$  return;
2  $r \leftarrow A[p]$ 
3  $i \leftarrow p - 1$ 
4  $j \leftarrow q + 1$ 
5 while  $i < j$  do
6     repeat
7          $i \leftarrow i + 1$ 
8     until  $A[i] \geq r$ 
9     repeat
10         $j \leftarrow j - 1$ 
11    until  $A[j] \leq r$ 
12    if  $i < j$ 
13        then поміняти  $A[i] \leftrightarrow A[j]$ 
14    Quicksort( $A, p, j$ )
15    Quicksort( $A, j + 1, q$ )
```

```
Partition( $A, p, q$ )
1  $x \leftarrow A[q]$ 
2  $i \leftarrow p - 1$ 
3 for  $j \leftarrow p$  to  $q - 1$ 
4 do if  $A[j] \leq x$ 
5     then
6          $i \leftarrow i + 1$ 
7         Swap  $A[i] \leftrightarrow A[j]$ 
8 Swap  $A[i + 1] \leftrightarrow A[q]$ 
8 return  $i + 1$ 
```

```
Quicksort( $A, p, q$ )
1 if  $p \geq q$  return;
2  $i \leftarrow$  Partition( $A, p, q$ )
3 Quicksort( $A, p, i - 1$ )
4 Quicksort( $A, i + 1, q$ )
```

# Реалізація на C++

Див Код