

## Сортировки

Теор (О сортировках сравнением)  $\forall g$

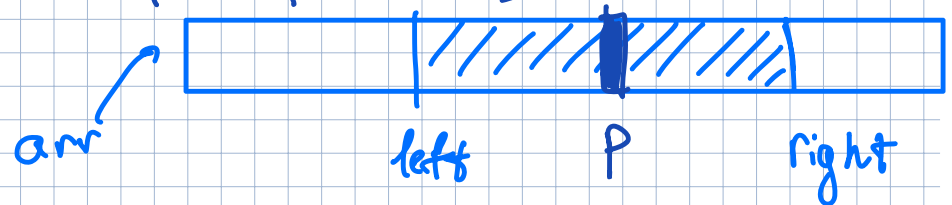
Любая сортировка сравнением  
имеет асимптотику  $\Omega(N \log N)$   $\rightarrow$  лучше, чем  $\Omega(N^2)$

if  $arr[i] > arr[j]$

сравнением

Q sort (Быстрая сортировка)

①



Функция  
?

Quick Sort (arr, left, right) ?  
if (right <= left) ?  
 $\rightarrow$  return

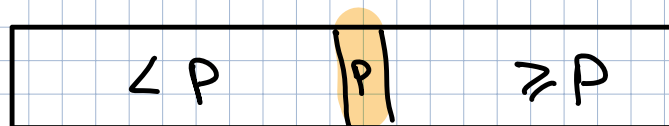
p = Partition (arr, left, right)

Quick Sort (arr, left, p-1) // рекурсия

Quick Sort (arr, p+1, right)

}

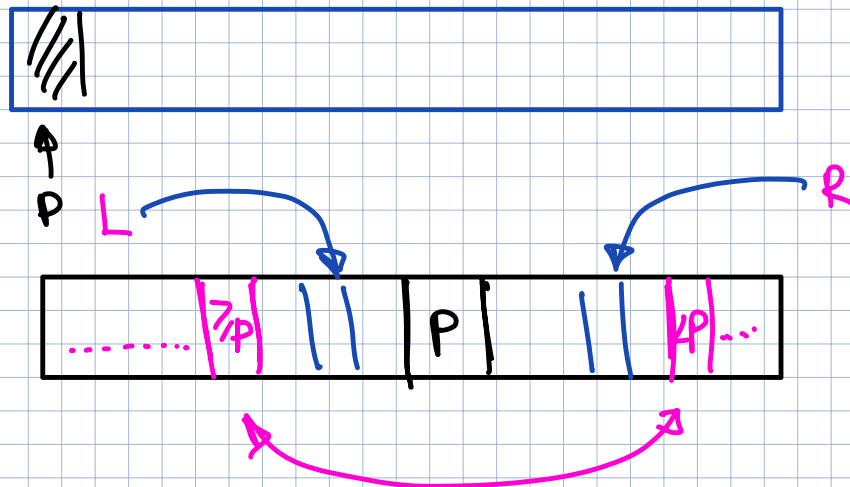
II) kono na komanu an Partition?



komanu  $O(N)$

III) Partition (krap)

pivot



```

① quicksort Partition (arr, left, right)
{
    pivot = arr[left]
    L = left
    R = right
    while (L < R)
    {
        while (arr[L] <= pivot)
            L++;
        while (arr[R] >= pivot)
            R--;
        if (L < R)
            SWAP (arr[L], arr[R])
    }
    return L
}

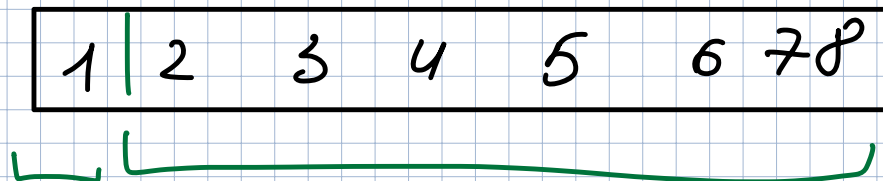
```

Partition



pivot = 5

IV) Нахудший случай



$N$  раз  $O(N)$   
 $\downarrow$   
 $O(N^2)$

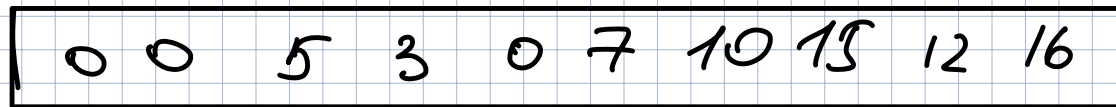
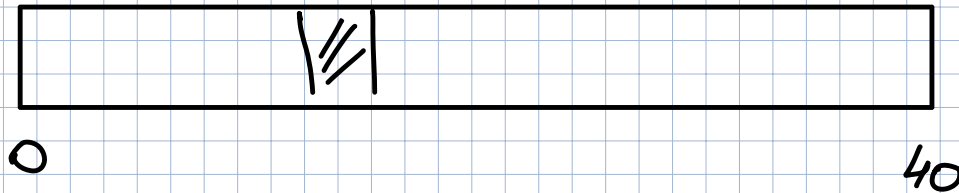
V) Выбор опорного (pivot)

- 1) arr[left]
- 2) arr[right]

3)  $\text{arr} \left[ \frac{\text{left} + \text{right}}{2} \right]$

4)  $\text{arr} [\text{rand}(\text{left}, \text{right})]$

Сортировка  
поделом



$N$

0	$\times 3$
1	$\times 0$
2	$\times 0$
3	$\times 1$
...	...

$\rightarrow$

0 0 0  
3  
0 0 3  
1 ...

$\rightarrow O(N \log N)$

Алгоритм:

$A = [0 \dots 0]$  - сколько есть  
элементов  $id$  со значением

$O(N)$

for  $i = 0 \dots N$ :

$A[arr[i]]++$ ;

$i = 0$

for  $j = 0 \dots \text{length}(A)$

{

while  $A[j] > 0$ :

$arr[i] = j$ ;

$A[j]--$ ;

$i++$

}

$O(N+K)$

$K$

Асимптотика:

$O(N+K)$

$K = O(N)$

$O(N)$

Устойчива? нет

# Counting Sort

python

Counting Sort (arr, N, k)

A = [0...0] greater k

B = [....] greater N - <sup>max</sup> <sub>< min</sub>

for i = 0... N:  
A[arr[i]] ++

for i = 1... k  
A[i] = A[i] + A[i-1]

for j = N.... 0 : arr[j]

B[A[arr[j]]] = arr[j]  
-- A[arr[j]]

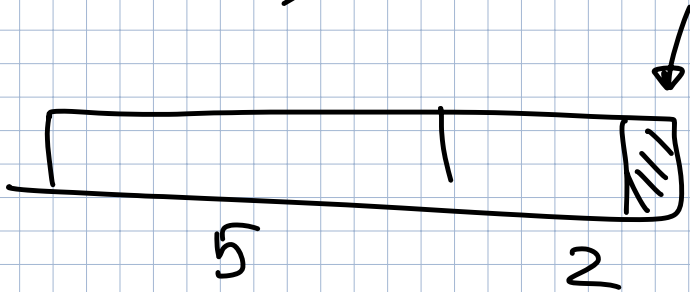
return B

3

$$A[0] = 5$$

$$A[1] = 2$$

$$A[1] = 7$$



Пример:

0 3 1 1 2 3 0 1 ~~2~~ ~~2~~ ~~3~~ ~~0~~

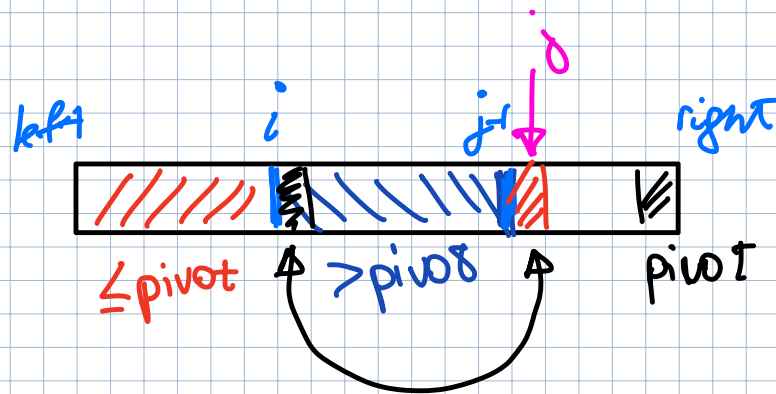
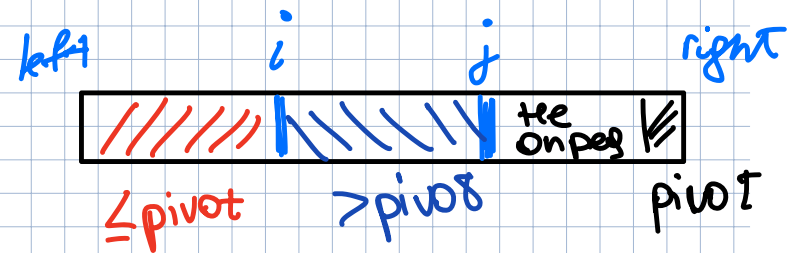
0	1	2	3	
3	4	3	3	← A

0	1	2	3	
2	6	8	12	← A

0	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	1	1	1	1	2	2	1	1		3



Разделение Помымо



Partition (arr, left, right)

3

pivot = arr[right]

i = left

for j = left ... (right - 1)

2

if arr[j]  $\leq$  pivot:

```
        } swap(arr[i], arr[j])  
        i++  
    }  
    }  
    swap(arr[i], arr[right])  
    return n (i+1)  
}
```