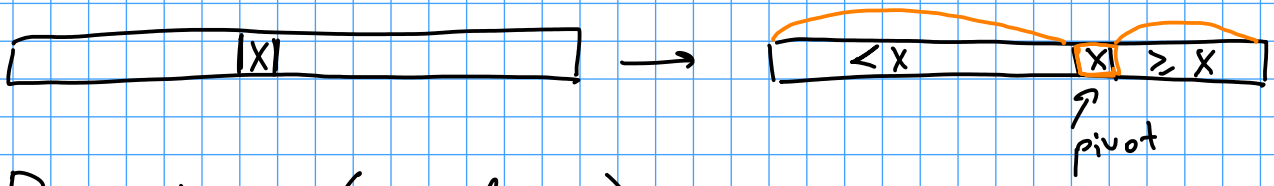
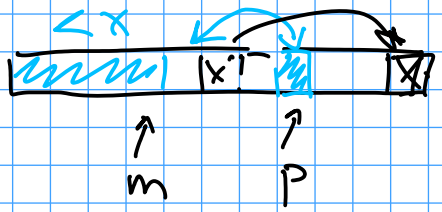
Быстрая сортировка

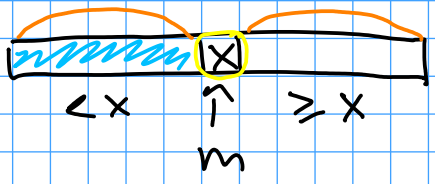
QuickSort

Идея




```
Partition(A, l, r)
    xi ← Pivot(A, l, r)
    A[xi] ↔ A[r]
    m = l
    for p = l to r-1:
        if A[p] ≤ A[r]
            A[p] ↔ A[m]
            m = m+1
    A[r] ↔ A[m]                    O(n)
    return m
```






```
QuickSort(A, l, r)
    m = Partition(A, l, r)
    QuickSort(A, l, m-1)
    QuickSort(A, m+1, r)
```

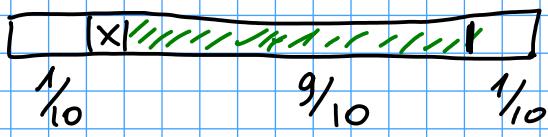
$T(n) = T(n-1) + T(0) + O(n)$

$T(n) = O\left(\sum_{k=1}^{n} k\right) = O(n^2)$

```
Pivot(A, l, r):
    return l
```

$T(n) = \Omega(n^2)$

Randomized Pivot (A, l, r)
return Random(l, r)



Глубина рекурсии:

$\log_{\frac{10}{9}} n = O(\log n)$

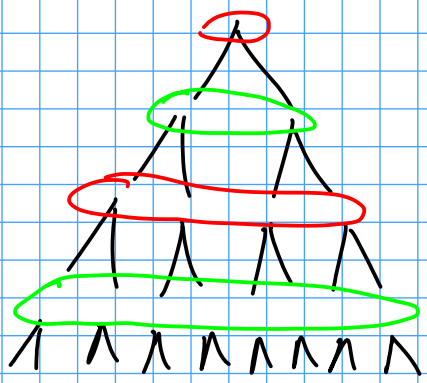
$T(n) = O(n \log n)$

Partition

$\left(\frac{4}{5}\right)^{\#} \to 0$

Орлом $\frac{1}{5}$
Решка $\frac{4}{5}$

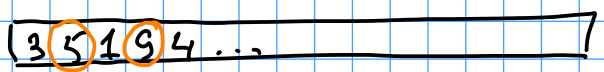
$\Rightarrow$ $\frac{1}{5}$ орлов $\frac{4}{5}$ решки



$T(n) \to T(n-1)$

$T(\frac{1}{10}n)$ $T(\frac{9}{10}n)$

$O(E[\#\text{сравнений}]) = E(T(n))$



$P(5 \text{ cmp } 9) - ?$

$i \ldots j$ $\quad \frac{1}{j-i+1}$

$P(i \text{ cmp } j) = \frac{2}{j-i+1}$

$E[\#\text{сравнений}] = \sum_{1 \le i < j \le n} P(i \text{ cmp } j) =$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} = \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \leq$$

$$\leq \sum_{i=1}^{n-1} \left(\sum_{k=1}^{n} \frac{2}{k+1} \right) = \sum_{i=1}^{n-1} O(\log n) =$$

$$= (n-1)\, O(\log n) = O(n \log n)$$

$$E(T(n)) = O(n \log n)$$

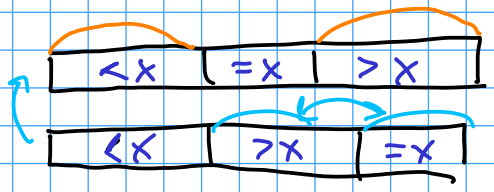
Алгоритм QuickSort работает $O(n \log n)$ в среднем.

$$\frac{n - \log n}{n} O(n \log n) + \frac{\log n}{n} O(n^2) \neq O(n \log n)$$

1 — 0

- IntroSort(A):
  QuickSort(A)
  Если глубина рекурсии $> c \cdot \log n$
  $\Rightarrow$ HeapSort(A)

- QuickSort3



- QuickSortRec(A, l, r)
  while (l < r)
  m = Partition(A, l, r)
  if m > (l+r)/2
  QuickSortRec(A, m+1, r)

r = m - 1

else

QuickSort Rec (A, l, m - 1)

l = m + 1