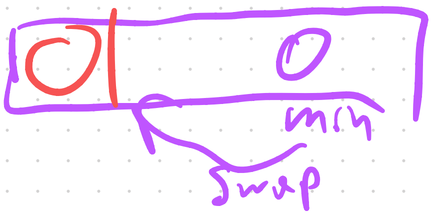
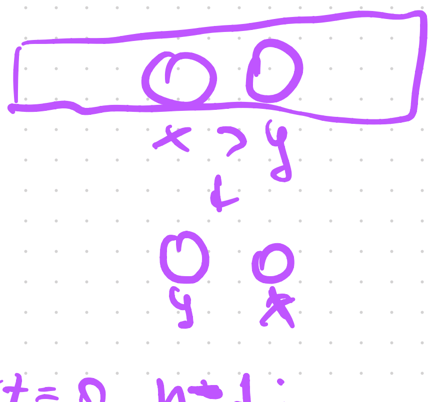
Квадратичные сортировки

- Selection sort

$\Theta(n^2)$

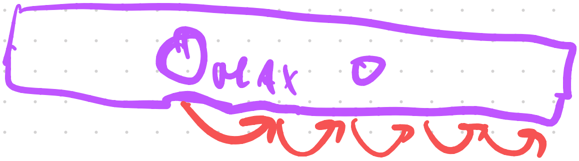


- Bubble Sort



$\Theta(n^2)$

```
for it = 0..n-1:
    for i = 0..n-2:
        if a_i > a_{i+1}:
            swap(a_i, a_{i+1})
```

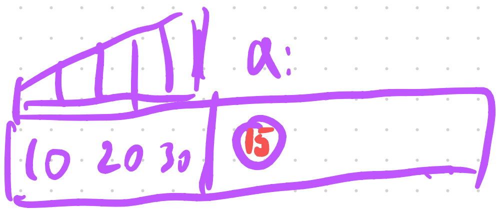


• Insertion Sort

1960
FRANK
LAZARUS

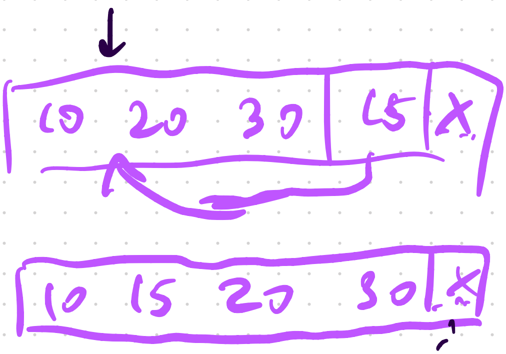
$f_1 = 100 n \log n$

$f_2 = 1 n^2$



пусть $[0, i)$ – уже отсортировано

$[0, i+1)$ отсортиров



For $i = 0 .. n-1$:

$j = i$

while $j > 0$ and $a_j < a_{j-1}$:

swap(a_j, a_{j-1})

$j--$

$\Theta(n^2)$

10 20 30 15

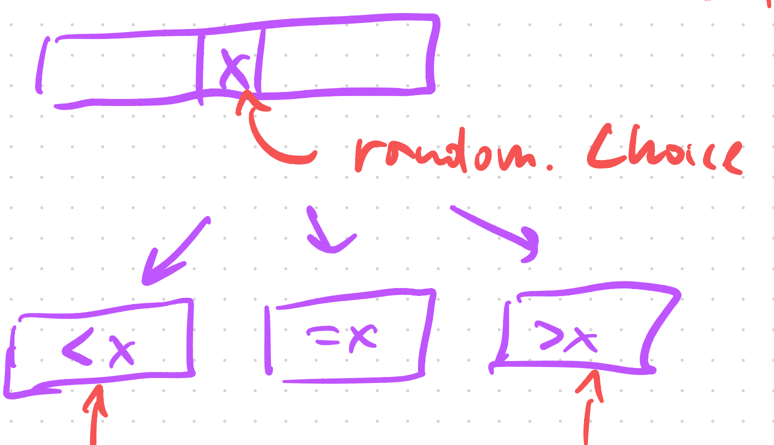
10 20 15 30

10 15 20 30 ✓

$n \lesssim 20$
50

Quick Sort

Hoare 1959 (1961)



Sort Sort

1 n-2

a n-1-a

$$T(n) \leq T(a) + T(n-1-a) + n$$

интуиция

$$a \approx n/2$$

$$\Rightarrow T(n) = \Theta(n \log n)$$

```
def sort(arr):
    if len(arr) <= 1:
        return arr

    x = random.choice(x)

    less, equal, greater = [], [], []

    for y in arr:
        if y < x:
            less.append(y)
        elif x < y:
            greater.append(y)
        else:
            equal.append(y)

    return sort(less) + equal + sort(greater)
```

(arr) Pivot

Оценка на время работы

Вероятность в алгоритмах

Probabilistic

работает с некоторой вероятностью

f, g

$f(x) \stackrel{?}{=} g(x)$

$x \in \mathbb{F}_p$

Randomized

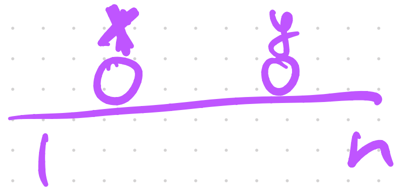
Алгоритм всегда выдаёт верный ответ, но время работы зависит от случайности ✓

Матожидание времени работы $= O(n \log n)$

Для простоты анализа, пусть мы сортируем перестановку

① Число сравнений

$a \leftarrow$ перестановка $\{1, \ldots, n\}$

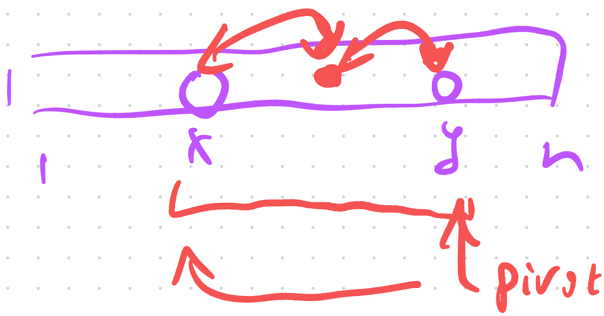


число сравнений =

$$= \sum_{1 \le x < y \le n} \mathbb{1}_{\substack{\text{сравнили} \\ x \text{ и } y}}$$

$$E[\text{число сравнений}] =$$

$$= \sum_{1 \le x < y \le n} \Pr\left[\substack{\text{сравнили} \\ x \text{ и } y}\right] =$$



$$= \sum_{1 \le x < y \le n} \frac{2}{y - x + 1}$$

$$= \sum_{len=2}^{n} \frac{2}{len} \cdot (n-len+1)$$

$$\leq \sum_{len=2}^{n} \frac{2}{len} \cdot n$$

$$= 2n \sum_{len=2}^{n} \frac{1}{len} =$$

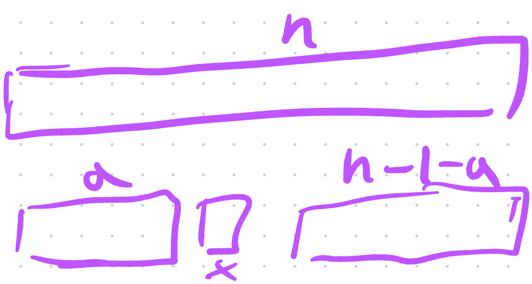
$$H_n = \sum_{k=1}^{n} 1/k \,.$$

$$H_n = \Theta(\log n)$$

$$H_n = \log n + \underbrace{\gamma}_{= 0.577} + o(1)$$

$$= 2n \log(n) + O(n)$$

D-60 2:



$$T(n) \leftarrow T(q) + T(n-1-q) + n$$

q — принимает значения от 0 до $n-1$ равновер.

$$T(n) = \frac{1}{n}\sum_{q=0}^{n-1} T(q) + T(n-1-q) + n$$

$$T(n) = \frac{2}{n}\sum_{q=0}^{n-1} T(q) + n$$

Цель: $T(n) = O(n \log n)$

$\exists C: T(n) \leqslant C n \log n$

Д-во по индукции:

База: $n = 0$

$T(0) = 0 \leqslant C \cdot \underbrace{0 \cdot \log 0}_{0}$

$n \leqslant n_0$ —||—

Переход:

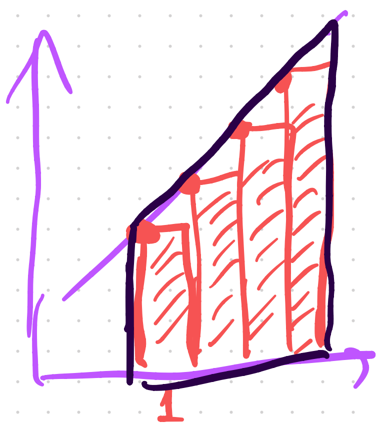
$$T(n) \leq Cn\log n$$
$$n \leq N-1$$

$$T(N) \ ? \ CN\log N$$

$$T(N) = \frac{2}{N}\sum_{a=0}^{N-1} T(a) + N$$

$$\leq \frac{2}{N}\sum_{a=1}^{N-1} Ca\log a + N$$

$$= \frac{2C}{N}\sum_{a=1}^{N-1} a\log a + N$$



$a\log a$

$$\le \frac{2c}{N}\int_1^N x\log x\,dx + N$$

$$= \frac{2c}{N}\left(\frac{N^2\log N}{2} - \frac{N^2}{4} + \frac{1}{4}\right) + N$$

$$= cN\log N \underbrace{- \frac{cN}{2} + \frac{c}{2N} + N}_{\le 0}$$

$-\gamma N$ $\qquad$ $T(N) \le cN\log N$

$$N\left(1 - \frac{c}{2}\right) + \frac{c}{2N} \le 0$$

① $c = 4.1$

② заметим что ф-ла будет верна с нек N_0

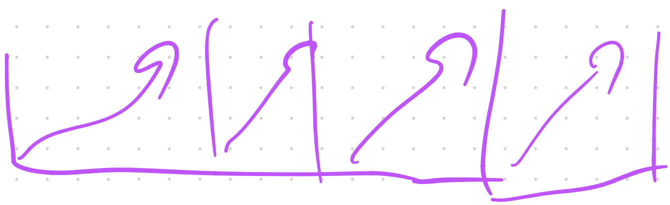
Язык	Sort
C++	Introsort
Python/Java	Timsort

mergesort

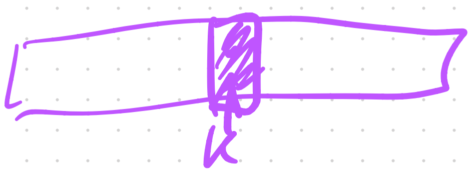
Introsort ← Q. Sort) $c \log N$

Heapsort) дальше

Insertion Sort



Порядковые статистики



Sort(a)

return a[k]

$k=0,\ k=n-1$

Q.Sort

Случайно

x

① $\underbrace{<x}_{\geq k+1}$ $=x$ $>x$

↑ k

② $\underbrace{<x \quad =x}_{\geq k+1}$ $>x$

③ $\underbrace{<x \quad =x \quad >x}_{\geq k+1}$

sort(arr, k)

```
def sort(arr):
    if len(arr) <= 1:
        return arr

    x = random.choice(x)

    less, equal, greater = [], [], []

    for y in arr:
        if y < x:
            less.append(y)
        elif x < y:
            greater.append(y)
        else:
            equal.append(y)

    return sort(less) + equal + [redacted](greater)
```

если $|less| \geq k+1$

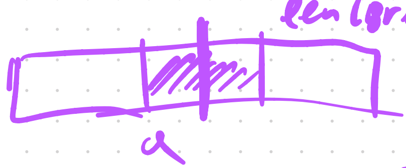
если $|less| + |equal| \leq k$

$|less| + |equal| + |Gr|$

$\geq k+1$

return less + equal + sort(greater, k − len(less) − len(equal))

время работы? len(less), len(greater) $\leq \frac{2n}{3}$



если в середину попали

$T(n) \leq n + T(\frac{2n}{3})$

$$T(n) \leq \frac{1}{3}\left(n + T\left(\frac{2n}{3}\right)\right) +$$

попали в серединку)

$$+ \frac{2}{3} T(n)$$

попали не в середину

$$\frac{1}{3} T(n) \leq \frac{1}{3}\left(n + T\left(\frac{2n}{3}\right)\right)$$

$$T(n) \leq n + T\left(\frac{2n}{3}\right)$$

$$\leq n + \frac{2n}{3} + \frac{4}{9}n + \dots$$

$$1 + q + q^2 + \dots q^n = \frac{1 - q^{n+1}}{1-q}$$

$$\leq 3n$$

Итб: $O(n)$ в среднем.

$\exists$ алг: $O(n)$ в худшем