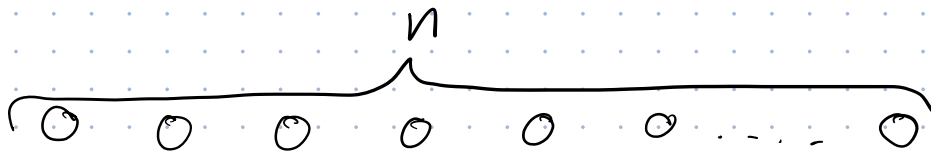


10.10.25

A3 N3.



$\{<, >, =\}$

$$3^k \geq n$$

$$k \geq \log_3 n$$

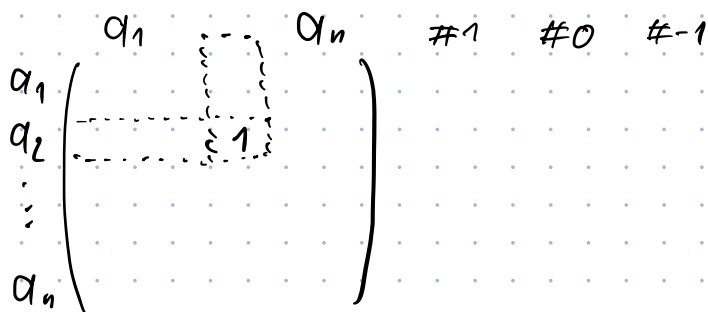
$$k = \lceil \log_3 n \rceil$$

СОРТИРОВКИ.

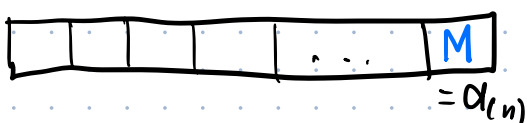
$[a_1, a_2, \dots, a_n]$



$[a_{j_1} \leq a_{j_2} \leq \dots \leq a_{j_n}]$



СОРТИРОВКА ПУЗЫРЬКОМ / bubble sort



def. ПОРЯДКОВАЯ СТАТИСТИКА

$a_{(k)}$ - ЭЛ-Т, КОТ. СТОИТ НА k -М МЕСТЕ В ОТСОРТ. МАССИВЕ

$$a_{(1)} = \min_i a_i; \quad a_{(n)} = \max_i a_i$$



АЛГ.

```
def bubble_sort(a):
```

```
    n = len(a)
```

```
    for i in range(n-1):
```

```
        max_ind = find_max_ind(a[0:n-i])
```

```
        swap(a, max_ind, n-i-1)
```

или $a[\text{max_ind}], a[n-i-1] =$
 $= a[n-i-1], a[\text{max_ind}]$

сложность:

$$\sum_{i=0}^{n-2} [n-i-1] = \Theta(n^2)$$

$$= \sum_{i=0}^{n-2} n - \sum_{i=0}^{n-2} i - \sum_{i=0}^{n-2} 1 =$$

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

$$= n(n-1) - \frac{(n-2)(n-1)}{2} - (n-1) =$$

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

$$= \frac{(n-1)n}{2} = \frac{n^2}{2} - \frac{n}{2} = \Theta(n^2)$$

```
def find_max_ind(a):
```

```
    max_ind = 0
```

```
    M = a[0]
```

```
    for i, el in enumerate(a[1:]):
```

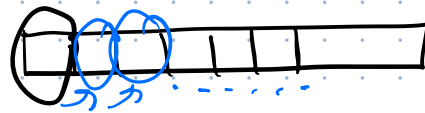
```
        if (el > M):
```

```
            M = el
```

```
            max_ind = i
```

```
    return max_ind
```

$a[0], \dots, a[n-i-1]$ — $n-i \geq 1$.



```
arr = ["a", "b", "c"]
```

```
for i, el in enumerate(arr):
```

```
    print(i, el)
```

```
def swap(a, i1, i2):
```

```
    if (i1 == i2):
```

```
        return
```

```
    tmp = a[i1]
```

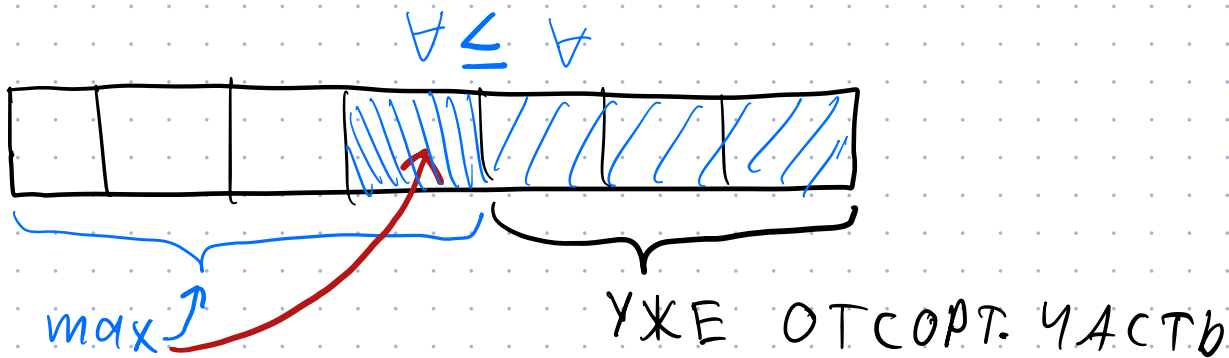
```
    a[i1] = a[i2]
```

```
    a[i2] = tmp
```

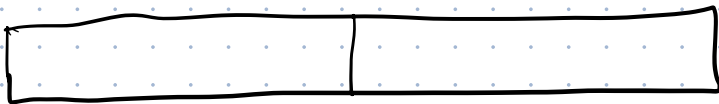
```
    a, b = b, a
```

ДОК-ВО КОРРЕКТНОСТИ

ПОСЛЕ k -ГО ШАГА ВНЕШ. ЦИКЛА ПОДМАССИВ
 $a[n-k-1:]$ ОТСОРТ. ПО ВОЗР.



СОРТИРОВКА СЛЯНИЕМ (merge sort)



```
def merge(arr1, arr2):
```

```
    i, j = 0, 0
```

```
    answer = []
```

```
    l1 = len(arr1)
```

```
    l2 = len(arr2)
```

```
    while (i < l1 or j < l2):
```

```
        if (i == l1):
```

```
            answer.append(arr2[j])
```

```
            i += 1 j += 1
```

```
        elif (j == l2):
```

```
            answer.append(arr1[i])
```

```
            i += 1
```

```
        else:
```

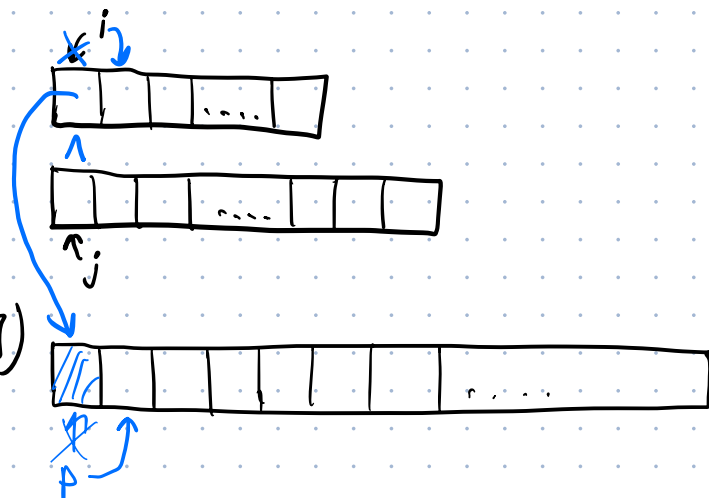
```
            if (arr1[i] > arr2[j]):
```

// ДЛИНА М И К СООТВ.

// ОБА УПОРЯД. ПО НЕУБ.

// ХОТИМ "СЛИТЬ" В ОДИН

// НЕУБ. МАССИВ



$\Theta(m + k)$

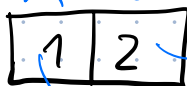
```

        answer.append(arr2[j])
        j += 1
    else:
        answer.append(arr1[i])
        i += 1

```

ПРИМЕР:

arr 1 =



arr 2 =



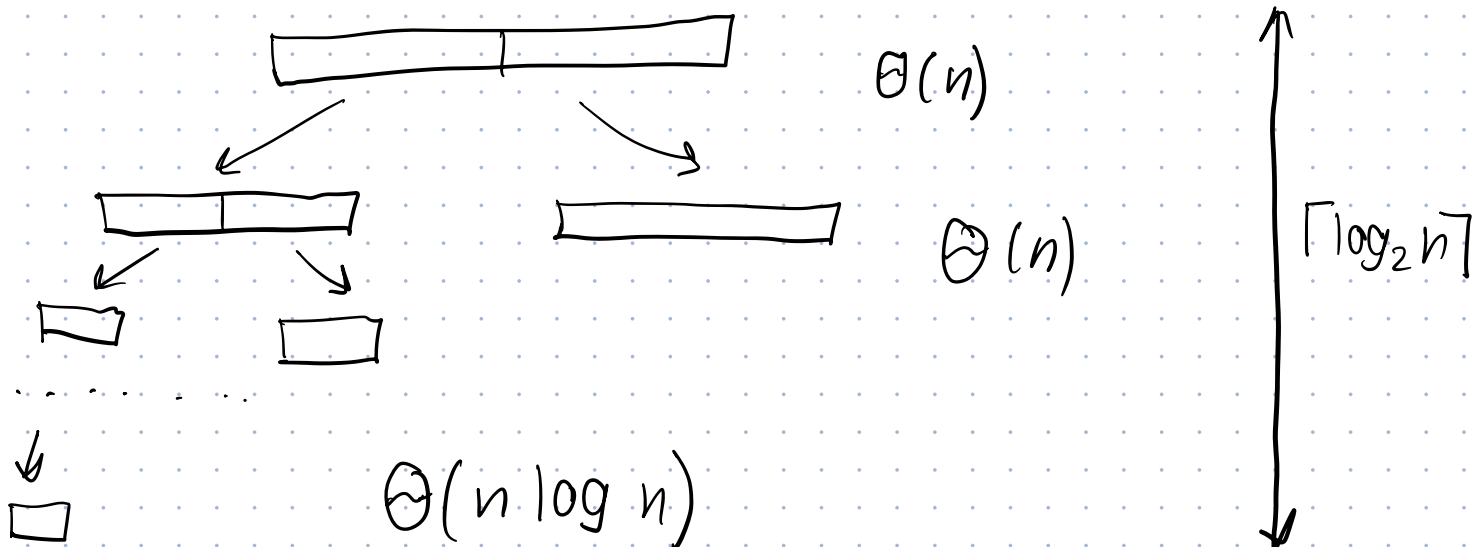
merge sort НА ОСНОВЕ merge

```

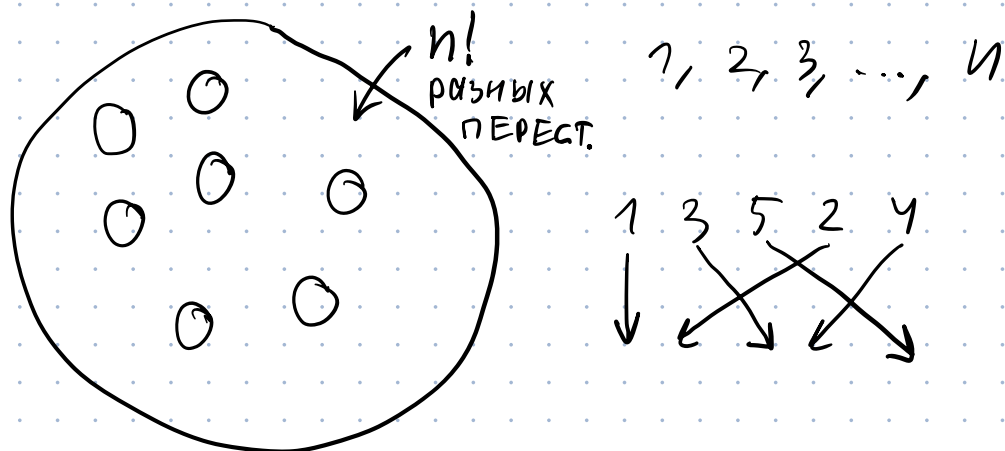
def merge_sort(a):
    if (len(a) <= 1):
        return a
    sorted_l = merge_sort(a[0: len(a)//2])
    sorted_r = merge_sort(a[len(a)//2: ])
    return merge(sorted_l, sorted_r)

```

СЛОЖНОСТЬ:



ОЦЕНКА СНИЗУ НА СЛОЖНОСТЬ СОРТИРОВОК СРАВНЕНИЯМИ
 МАСС. ИЗ n РАЗН. ЭЛ.



2 ОТВЕТА: $>$ $<$

КОЛ-ВО СРАВН. = КОЛ-ВО БИТ В КОДИР. ПЕРЕСТ.

k - НЕОБХ. ЧИСЛО СРАВН.

$$2^k \geq n!$$

$$k \geq \log_2(n!)$$

$$k = \lceil \log_2 n! \rceil$$

1 2 3
 1 3 2
 2 1 3
 2 3 1
 3 1 2
 3 2 1

$k=1:$

0
 1

$k=2:$

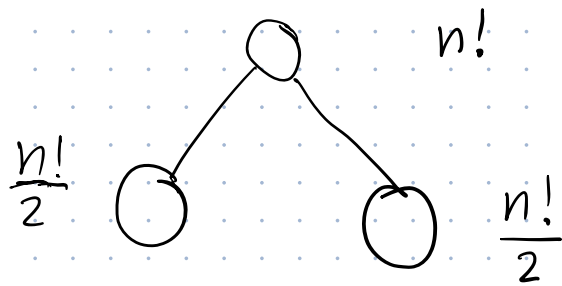
00
 01
 10
 11

$$\log_2 n! = \log_2 (1 \cdot 2 \cdot 3 \cdot \dots \cdot n) = \sum_{i=1}^n \log_2 i \leq$$

$$\leq \sum_{i=1}^n \log_2 n = n \log_2 n = O(n \log n)$$

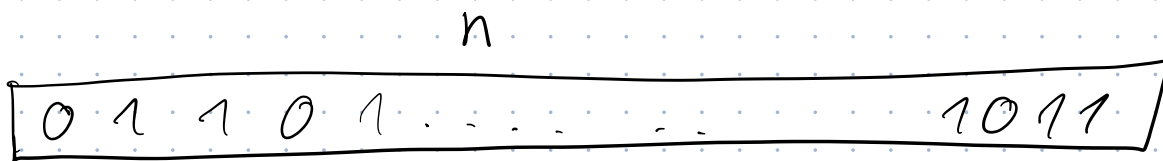
$$\geq \sum_{i=\frac{n}{2}}^n \log_2 i \geq \sum_{i=\frac{n}{2}}^n \log_2 \frac{n}{2} = \frac{n}{2} \log_2 \frac{n}{2} = \frac{1}{2} n (\log_2 n - \log_2 2) =$$

$$= \frac{1}{2} n \log_2 n - \frac{n}{2} = \Omega(n \log n)$$



$$\geq \frac{n!}{4}$$

$$\frac{n!}{2^n}$$



- 1) СЧИТАЕМ КОЛ-ВО НУЛЕЙ (ПУСТЬ ИХ К)
- 2) ПИШЕМ В ОТВ. К 0
- 3) ПИШ. $n-k$ 1

$$\Theta(n)$$

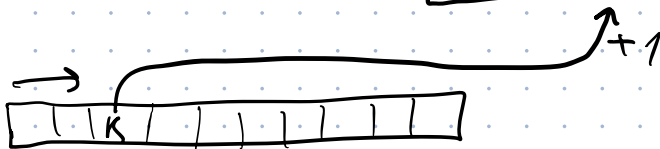
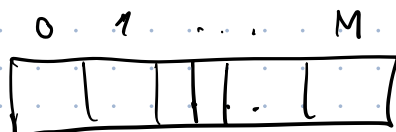
СОРТ. ПОДСЧЁТОМ (counting sort)

$$a_1, \dots, a_n$$

$$a_i \in [0, M] \cap \mathbb{Z}$$

$$M = 2^n$$

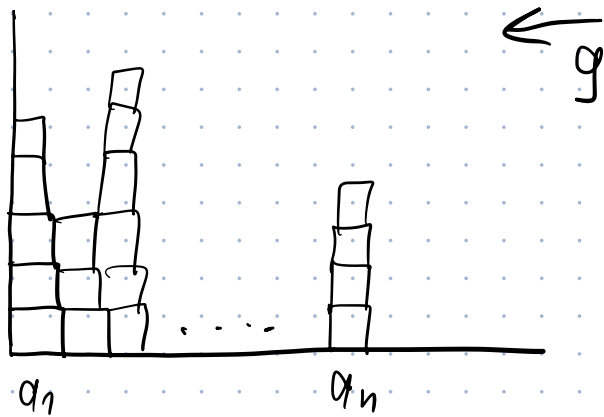
"МАССИВ ЧАСТОТ":



$$\Theta(n + M) \text{ ВРЕМ.}$$

$$\Theta(M) \text{ ПАМ.}$$

a_1, a_2, \dots, a_n



$$S = \frac{gt^2}{2} = w_k \cdot h$$

$$t = \sqrt{\frac{2 \cdot w_k \cdot h}{g}} = C \cdot \sqrt{h}$$