10.02.25

TEOP. A32

1.4. $T(n) = T(n-1) + 3n = T(n-2) + 3(n-1) + 3n$ (III)

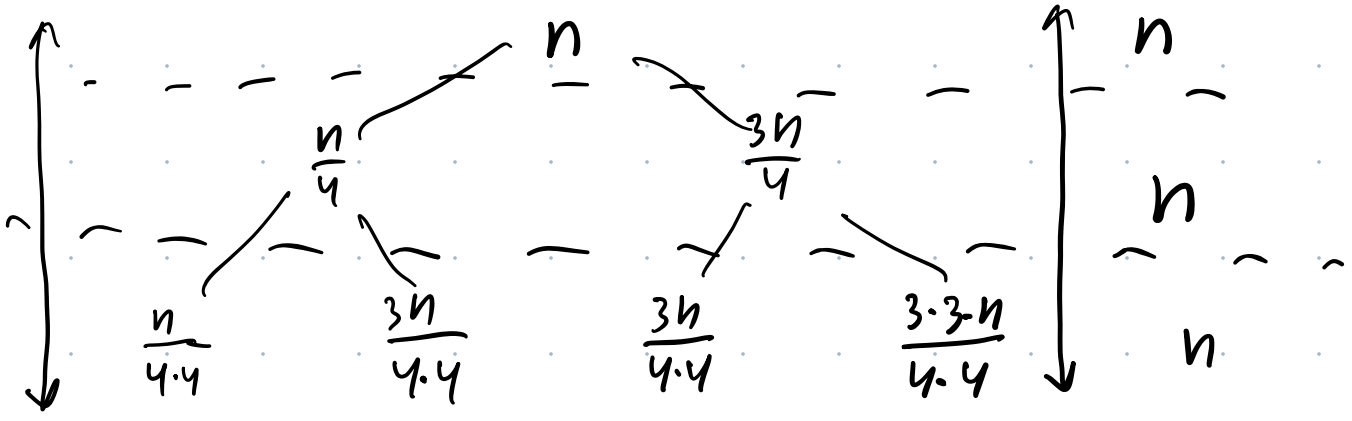
$T(1) = C$

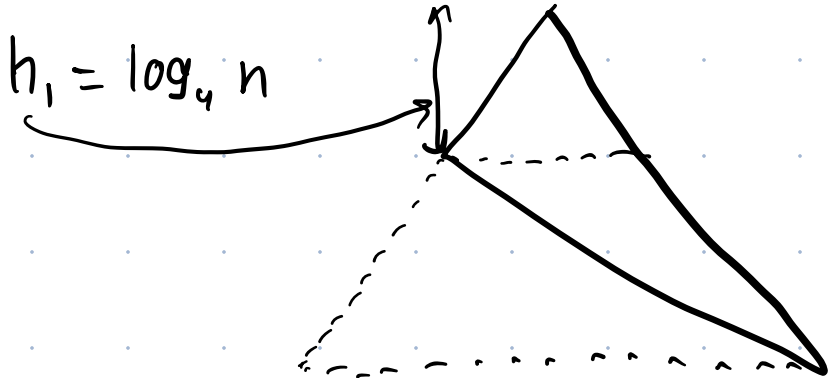
(III) $\sum_{i=0}^{n-2} 3(n-i) + C = 3\sum_{i=2}^{n} i + C = 3\left[\frac{n(n+1)}{2} - 3\right] + C = \Theta(n^2)$

1.5. $T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n$

$\geq 2T\left(\frac{n}{4}\right) + n$; $c = \log_4 2 = \frac{1}{2} < 1$; $T(n) = \Theta(n)$ ✗

$\leq 2T\left(\frac{3n}{4}\right) + n$; $c = \log_{\frac{4}{3}} 2 > 1$; $T(n) = \Theta\left(n^{\log_{\frac{4}{3}} 2}\right)$

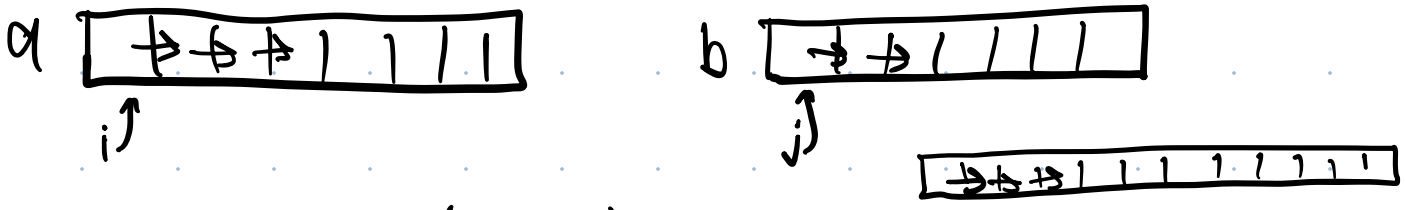




$h_r = \log_{\frac{4}{3}} n$

$$n \log_4 n \leq T(n) \leq n \log_{\frac{4}{3}} n$$

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




```
def merge(a, b):
    len_a = len(a)        # K
    len_b = len(b)        # m
    i, j = 0, 0
    merged = []

    while (i != len_a or j != len_b):
        if (i == len_a):
            merged.append(b[j])
            j += 1
        elif (j == len_b):
            merged.append(a[i])
            i += 1
        else:
            if (a[i] > b[j]):
                merged.append(b[j])
                j += 1
            else:
                merged.append(a[i])
                i += 1
```

$\Theta(K + m)$

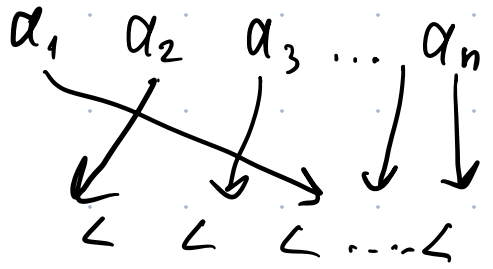
return merged

Нижняя оц. на сорт. сравнениями

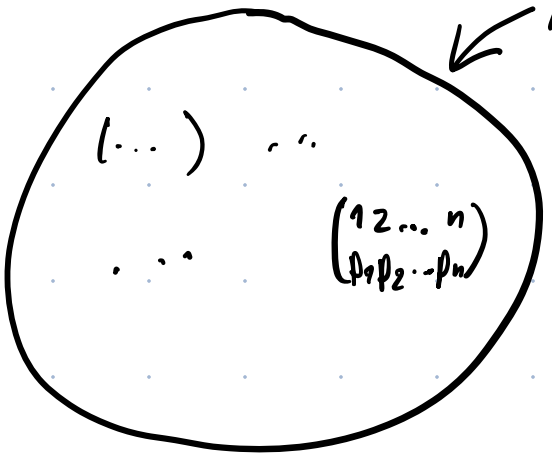
n чисел: 1, 2, ..., n

O(1) a ? b (какой эл-т больше)

различных перестановок n! штук



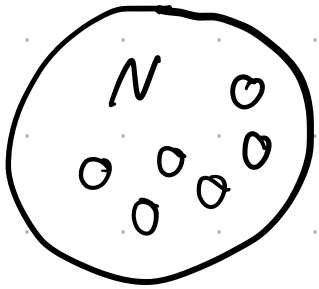
Для каждой перестановки ∃! обратная

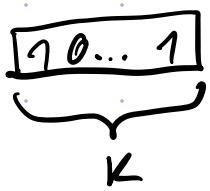


мн-во всех перестановок

n! элементов = N

длина кода $\geq \lceil \log_2 N \rceil$





$2^k \geq N$

$\log_2 2^k \geq \log_2 N$

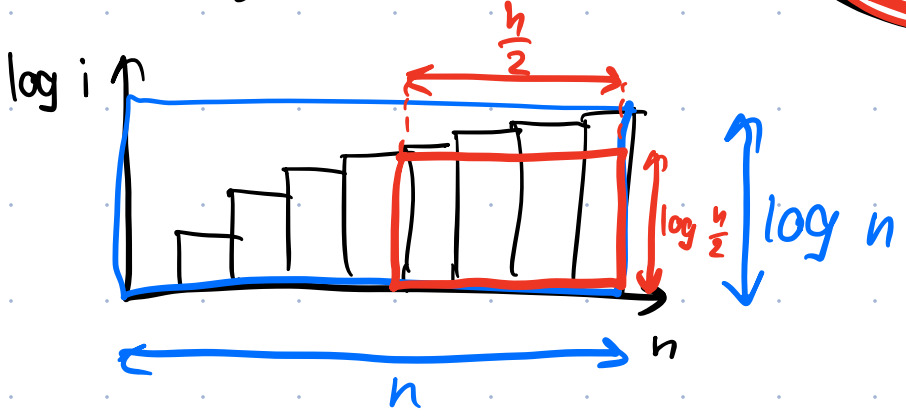
$k \geq \lceil \log_2 N \rceil$

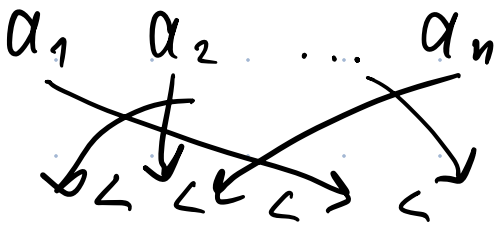
$$\log_2 N = \log_2 n! = \sum_{i=1}^{n} \log_2 i \leq \sum_{i=1}^{n} \log_2 n = n \log_2 n \simeq O(n \log n)$$

$$\log a \cdot b = \log a + \log b$$

$$\geq \sum_{i=\frac{n}{2}}^{n} \log_2 i \geq$$

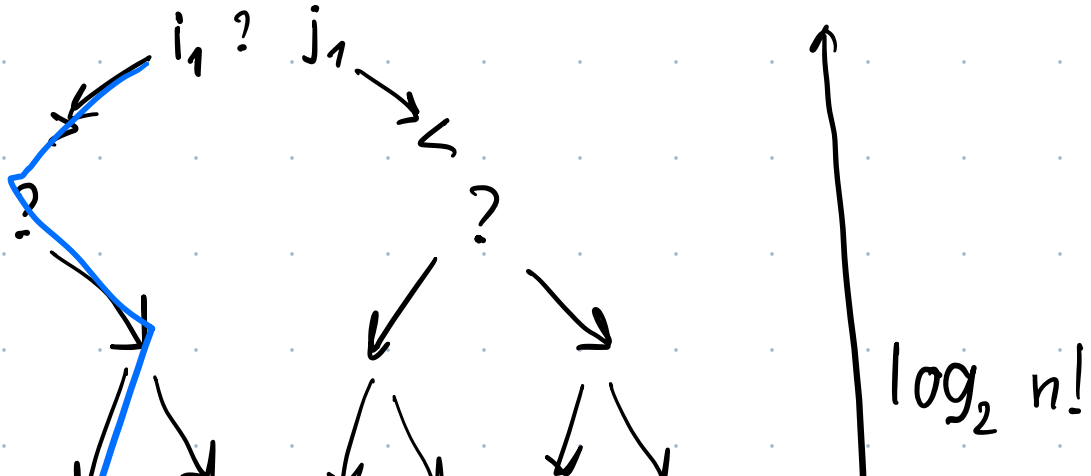
$$\geq \sum_{i=\frac{n}{2}}^{n} \log_2 \frac{n}{2} = \frac{n}{2} \log_2 \frac{n}{2} = \Omega(n \log n)$$

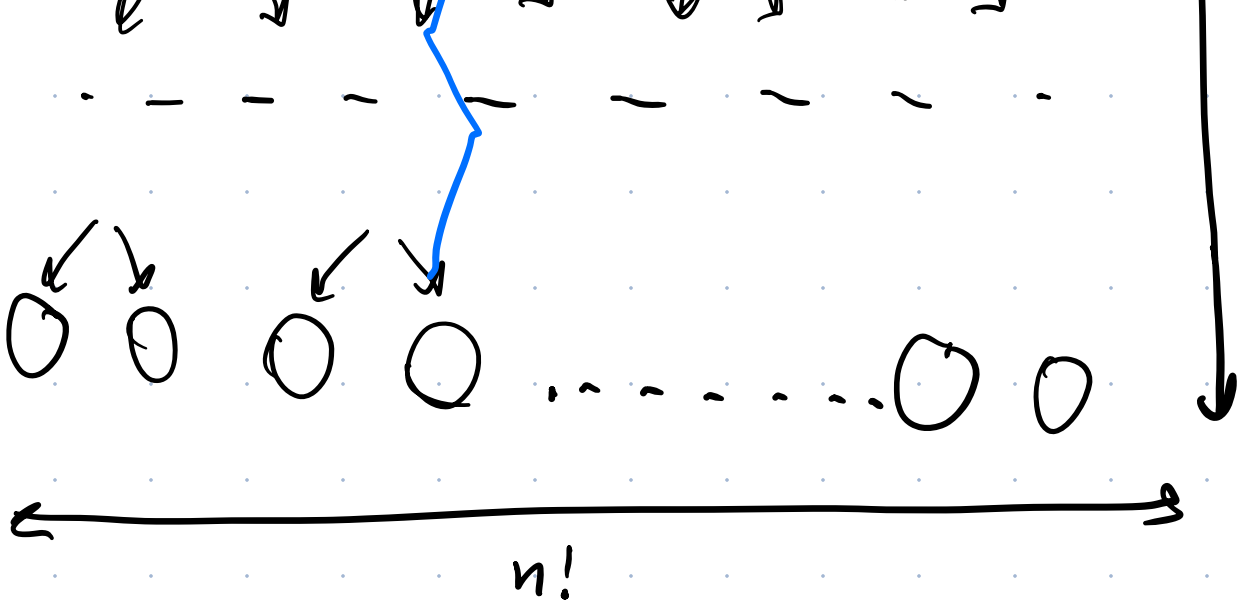




$\forall i \quad \exists j$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \dots & n \\ p_1 & p_2 & p_3 & & & p_n \end{pmatrix}$$






```
sorted_arr = sorted(arr, key = lambda i: i[0])
                                          i[1]
```

arr = [(1, 3), (5, 8), (5, -1), ...]

Сортировки не только сравн.

$a = [1, 1, 0, 1, \ldots, 1, 0]$ (n)

0 0 0 0 0 0 1
zeros = 5

```
zeros = 0
for i in range(len(a)):
    if (a[i] == 0):
        zeros += 1
```

```
for i in range(zeros):
    a[i] = 0
```

```
for i in range(zeros+1, len(a)):
    a[i] = 1
```

Сортировка подсчётом (counting sort)

$\Theta(n)$ по времени

$\Theta(1)$ по памяти

ОБОБЩИМ counting НА int ОТ 0 ДО M

```
f = [0 for _ in range(M+1)]

for i in range(len(a)):
    if (a[i] == 0):
        zeros += 1
    f[a[i]] += 1
ind = 0
for i in range(M+1):
    for j in range(f[i]):
        a[ind] = i
        ind += 1
```

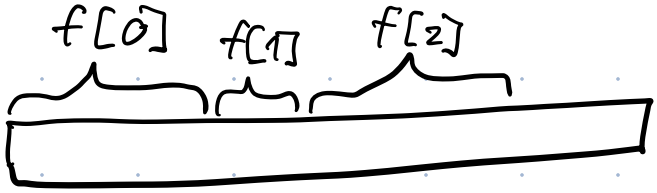
M = 3

0 0 1 3 3 0 0

f = [4 1 0 2]

0 1 2 3

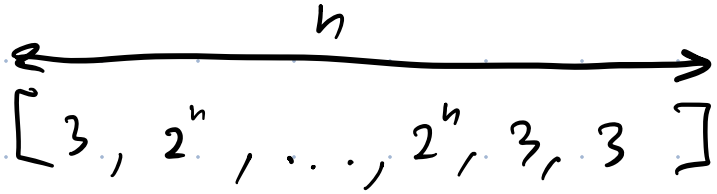
0 0 0 0 1 3 3



ВРЕМЯ: $\Theta(n+M)$

ПАМЯТЬ: $\Theta(M)$

КОНТРПРИМЕР:



$\Theta(n+2^n)$