Нижняя оценка на сортировки

Можно или нельзя отсортировать данные за $o(n \log n)$, то

да*

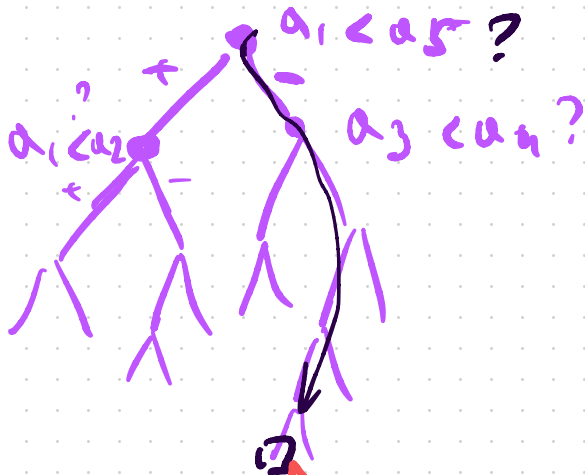
Th Пусть Алгоритм сортировки использует только сравнения $\Rightarrow$ то он работает $\Omega(n \log n)$

Д-во:

$[1\ 2\ 3\ \ldots\ n]$ — $n!$

↑ пусть перестановки

$a_i \overset{?}{<} a_j$



$$\text{leaves} \geq n!$$

$$\text{MAX depth} \geq \log_2(n!)$$

$$= \log_2 e \cdot \ln(n!)$$

$$= \underbrace{\log_2 e \,(n \log n + O(n))}$$

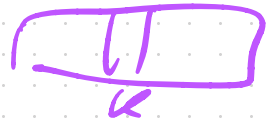
Кол-во сколько сравнений

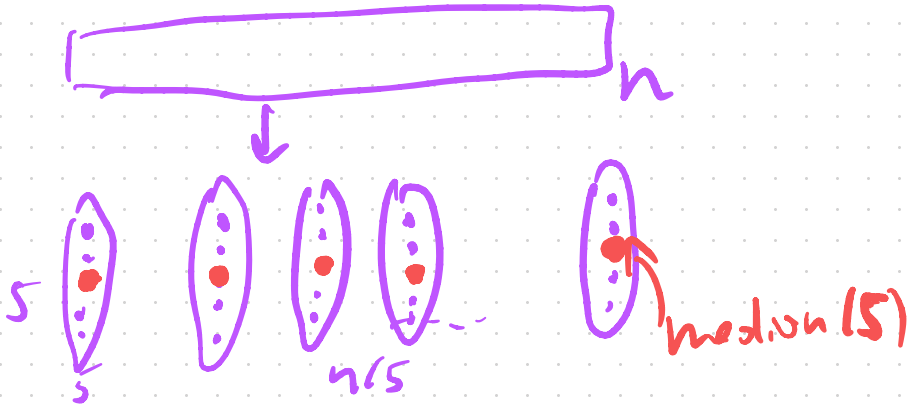
□

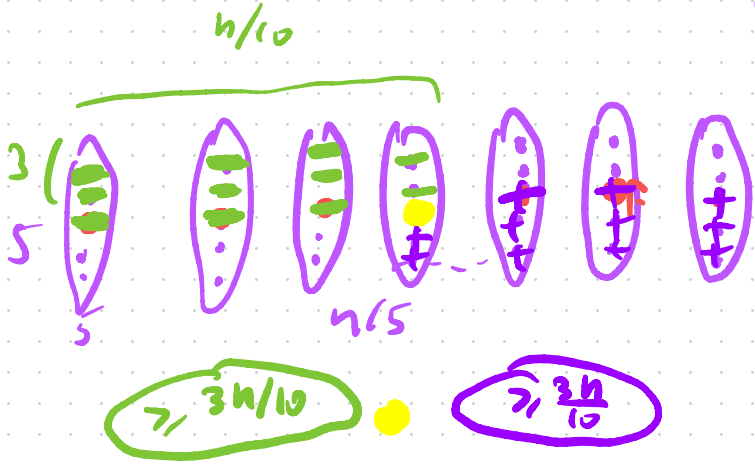
Замечание: Q. Sort: $2n \log n + O(n)$

Замечание: Э-во в случае рандомиз. Алгоритма

Median of Medians



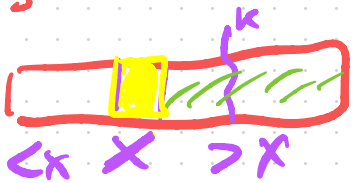


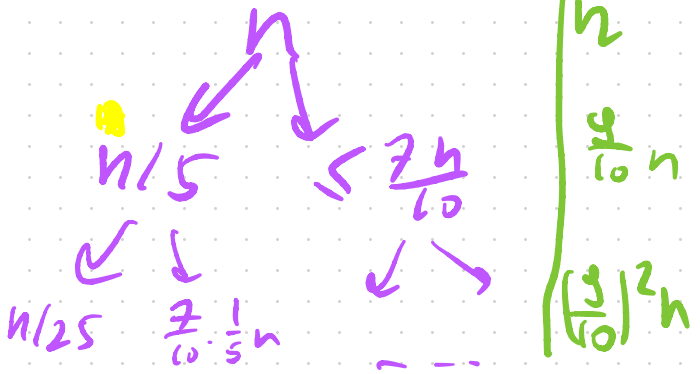


1) Найти ● – median of medians

2) сделать по нему partition

3) "односторонний QSORT"





$$T(n) = \frac{n}{5} \cdot \underbrace{6}_{O(1)} + n + T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right)$$

$$= \frac{11}{5} n + T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right)$$

$$= \frac{11}{5} (\Sigma n_i) =$$

$$= \frac{11}{5}\left(n + \frac{n}{5} + \frac{3n}{10} + \underbrace{\frac{n}{2} + \cdots}_{11 \text{ on } n}\right)$$

$$= 22n$$

Count Sort

$a_1 \dots a_n$ $\quad a_i \in [0; m)$

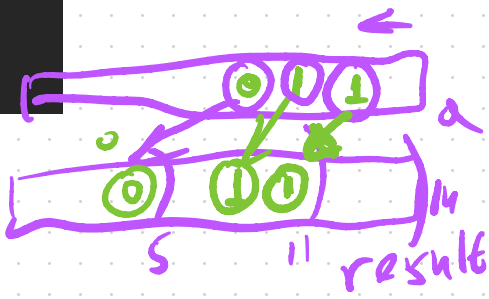
целые числа.

```python
def sort(a, m):
    count = [0 for _ in range(m)]

    for x in a:
        count[x] += 1

    result = []
    for (i, num) in enumerate(count):
        for _ in range(num):
            result.append(i)



def stable_sort(a, m):
    count = [0 for _ in range(m)]

    for x in a:
        count[x] += 1

    for i in range(1, m):
        count[i] += count[i - 1]

    result = [None for _ in range(len(a))]
    for val in reversed(a):
        count[val] -= 1
        result[count[val]] = val
```

$O(n+m)$

0 1 2
5 6 3

5 11 14



Сортировка пар

(x, y)

(z, w)

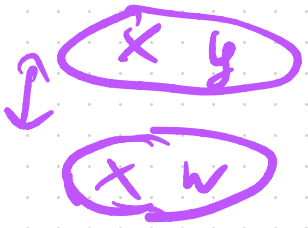
↑ ↑

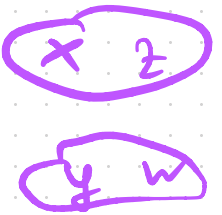
[0, m) - целые числа

Stable_counting_sort

сначала по второму элементу пары,

а потом по первому



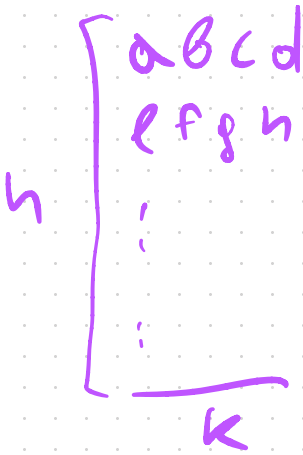


2) нет

1) первый элемент одинаковый

$O(n+m)$

Dict Sort



$O(k \cdot (n + |\Sigma|))$

Жадность

Выполнить задачи в порядке $d_i \uparrow$

Can I S) – правда ли, что можно выполнить ми-во S.

$Can(ALL) = OR_{last \in ALL}$

$Can(ALL \setminus \{last\})$

and

$time(ALL \setminus \{last\}) + time(\{last\}) \leq d(\{last\})$

$time(ALL)$

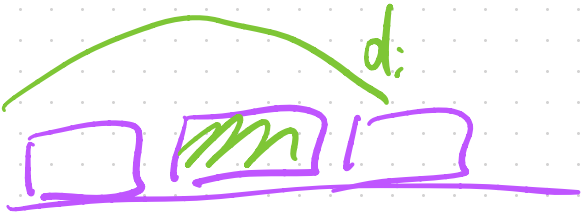
$Can(ALL) = OR_{last \in ALL} \leq 1$

1

$Can(ALL \setminus \{last\})$

and $time(ALL) \leq d_{last}$

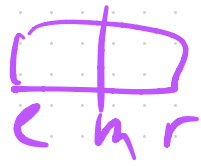
$last : d_{last} \to MAX$



Merge Sort

Не рекурсивно и с эффективным исп. памятью

```python
def merge_sort(arr, l, r):
    # отсортировать на месте arr[l...r-1]
    if r - l <= 1:
        return

    m = (l + r) // 2
    merge_sort(l, m)
    merge_sort(m, r)

    result = []
    pA, pB = l, m

    while pA < m and pB < r:
        if arr[pA] <= arr[pB]:
            result.append(arr[pA])
            pA += 1
        else:
            result.append(arr[pB])
            pB += 1

    while pA < m:
        result.append(arr[pA])
        pA += 1

    while pB < r:
        result.append(arr[pB])
        pB += 1

    # copy result to arr[l..r-1]
```



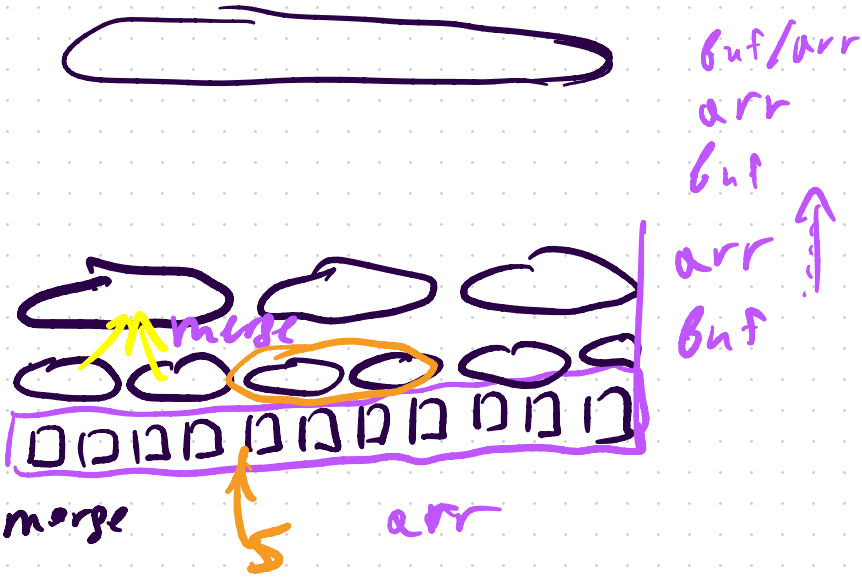
[l, m), [m, r)

merge(l, m, r)

← Result

один раз

можно выделять



for $w \in [1, 2, 4, 8 \ldots]$

for $s = [0, 2w, 4w \ldots]$

merge$(s, s+w, s+2w)$

$\min(s+w, n)$ $\min(s+2w, n)$