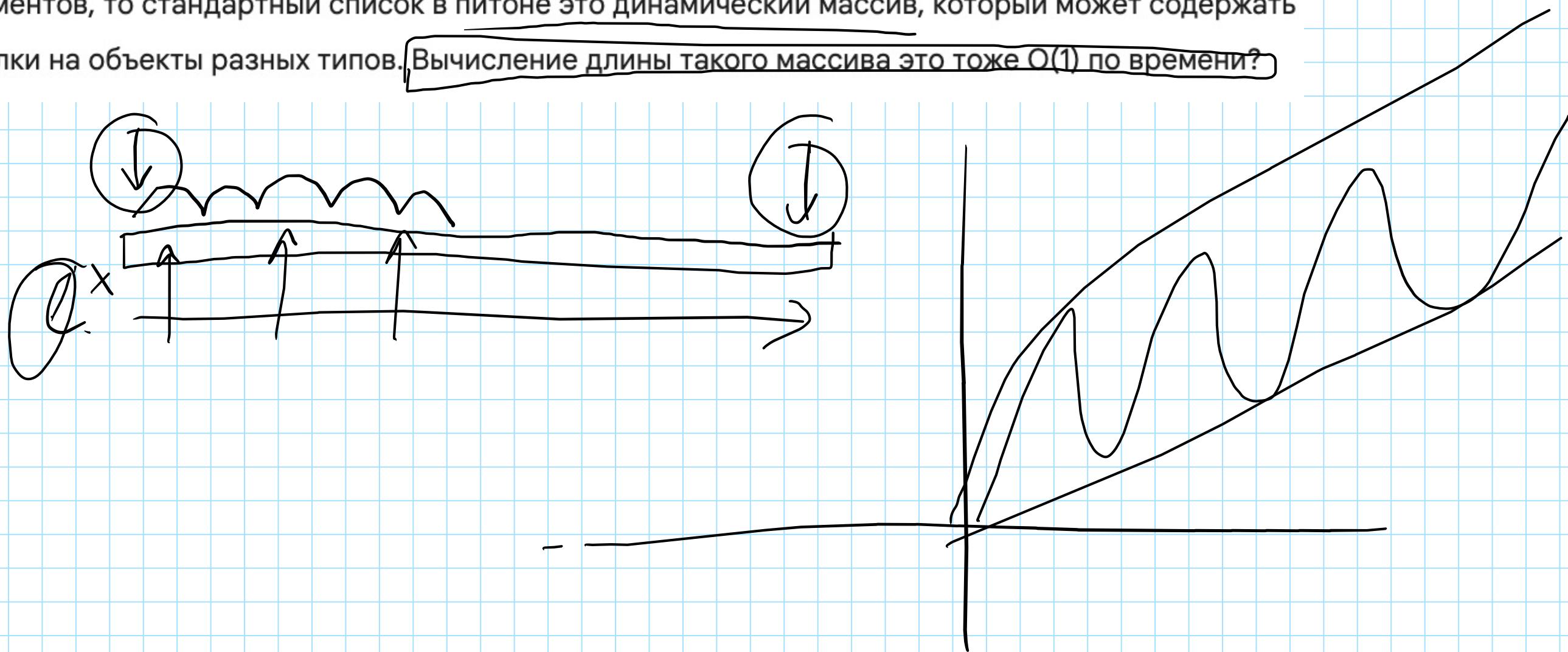


Как нам убедиться в том что найденная наилучшая оценка сверху(или снизу) действительно наилучшая? Как доказать что лучше оценки нет?

$O(n^k)$

$$O(\cdot) = O(\underline{\cdot}) \Rightarrow O(\rightarrow)$$

Объясните пожалуйста, почему определение длины массива это $O(1)$ по времени. Если в языках со строгой типизацией типа c++ при инициализации массива заранее задается размер и тип элементов, то стандартный список в питоне это динамический массив, который может содержать ссылки на объекты разных типов. Вычисление длины такого массива это тоже $O(1)$ по времени?



$\text{len}(s) = n$

6. Палиндромом называется строка, которая одинаково читается слева направо и справа налево. Рассмотрим некоторую строку S , состоящую из маленьких латинских букв. Какое минимальное количество букв (возможно нуль) нужно в ней удалить, чтобы она не являлась палиндромом (если это возможно)? Предложите алгоритм решения задачи, докажите его корректность, оцените сложность.

1) $abba$ — пал.

2) abb — не пал.

3) a — пал

4)

$\overbrace{aaa \dots a}^n \quad \overbrace{b \dots b}^n \quad \overbrace{a \dots e}^n$

$\rightarrow l$

Ответ

1

0

∞

$\leftarrow l$
 $\leftarrow n$
 $\Rightarrow O(n)$

① Не пал \rightarrow Ответ 0

② Пал \rightarrow $l = \underbrace{l \dots l}_n \wedge l \rightarrow$ Ответ ∞

③ Пал \wedge хот есть хотя бы 2 раз. буквы

\downarrow
Ответ "1"

хотеть a, b
 $i+1$

$\overbrace{aa \dots a}^i b ?$

$b \underbrace{aa \dots a}_i$

$i-i$ слева : b
 $i-i$ справа : a

b

a

</

3. На вход задачи поступают массив $a = [a_1, \dots, a_n]$ и число S . Требуется проверить, есть ли в массиве элементы a_i и a_j ($i \neq j$), такие что $a_i + a_j = S$.

1. Постройте алгоритм, решающий задачу за $O(n \log n)$.

2. Представьте, что у вас есть структура данных «множество», добавление элементов в которую стоит $O(1)$ и проверка наличия элемента в структуре так же стоит $O(1)$. Решите задачу, используя эту структуру за $O(n)$.

3. Постройте алгоритм, решающий задачу за линейное время, при условии, что массив a отсортирован, используя дополнительно $O(1)$ памяти.

$1 \in \{5, 0, -1\}$, $S = 4 \rightarrow \text{нет}$
 For $i = 1 \dots n$, $S = 8 \rightarrow \text{нет}$
 For $j = i+1 \dots n$, $n-1, n-2, n-3, \dots$
 $O(n^2) - \text{трабв. } n + n + n + \dots = n^2$

④ $O(n \log n) \rightarrow \text{sort}$



$\ell = 1$
 $r = n$
 $a[\ell] < a[r] > S$

$a[\ell] + a[r] < S$
 $\ell += 1$

$a[\ell] + a[r] == S$
 return Yes

return No

3. На вход задачи поступают массив $a = [a_1, \dots, a_n]$ и число S . Требуется проверить, есть ли в массиве элементы a_i и a_j ($i \neq j$), такие что $a_i + a_j = S$.

1. Постройте алгоритм, решающий задачу за $O(n \log n)$.

2. Представьте, что у вас есть структура данных «множество», добавление элементов в которую стоит $O(1)$ и проверка наличия элемента в структуре так же стоит $O(1)$. Решите задачу, используя эту структуру за $O(n)$.

3. Постройте алгоритм, решающий задачу за линейное время, при условии, что массив a отсортирован, используя дополнительно $O(1)$ памяти.

$O(n)$

$B = \{a_1, \dots, a_n\}$
 for $i = 1 \dots n$
 $\rightarrow \text{if } S - a_i \in B \text{ then } O(n)$
 return Yes

return No

$S = 2$

$S - a_1 = 1$
 $B = \{y\}$

for $i = 1 \dots n$:
 $\text{if } S - a_i \in B \text{ then }$
 return Yes
 $B.add(a_i)$

return No

1
 $a_1 \dots a_n$

$O(n)$

$S - a_i \dots S - a_n$

$a_i + a_j = S$

$\exists a_j = S - a_i$

$1 \in \{5, 0, -1\}$

$S = 4 \rightarrow \text{нет}$

$S = 8 \rightarrow \text{нет}$

$n-1, n-2, n-3, \dots$

$O(n^2) - \text{трабв. } n + n + n + \dots = n^2$

sort

$\ell \nabla$

$r \nabla$

$1 \dots 100500$

$l \nabla$

$r \nabla$

$1 \dots 10$

5. Дан массив a из n положительных целых чисел. Нужно разбить этот массив на максимальное количество непрерывных подмассивов так, чтобы после сортировки элементов внутри каждого подмассива весь массив стал отсортированным.

Более формально, нужно выбрать индексы i_1, \dots, i_{k-1} ($i_k = n$), такие что массив разбивается на подмассивы

$$[a_1, \dots, a_{i_1}], [a_{i_1+1}, \dots, a_{i_2}], \dots, [a_{i_{k-1}+1}, \dots, a_{i_k}],$$

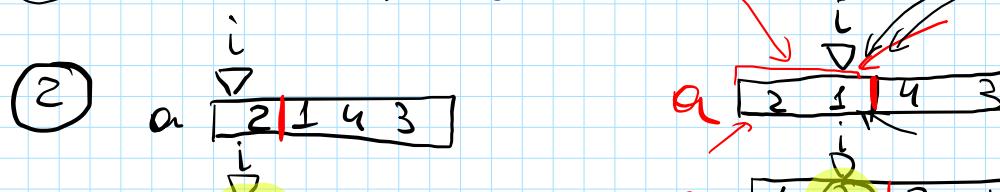
в результате сортировки каждого из которых (и обратной склейки) получается отсортированный исходный массив; количество блоков разбиения k должно быть максимально возможным.

Предложите $O(n \log n)$ алгоритм для решения этой задачи, можно использовать $O(n)$ дополнительной памяти.

Обязательно обоснуйте работоспособность вашего алгоритма. Оцените его сложность и объём необходимой дополнительной памяти. Мы будем благодарны, если описание алгоритма будет дано не листингом кода, а просто текстом, возможно, с привлечением псевкодода.

$$\begin{aligned} a &: 2143 \rightarrow 12|34 \rightarrow 1234 & (2) \\ 1234 &\rightarrow 1|2|3|4 \rightarrow 1234 & (5) \\ 4321 &\rightarrow 14321 \rightarrow 1234 \\ a & 2143 \\ s & 1234 \end{aligned}$$

$$(1) s = \text{sort}(a) \quad O(n \log n)$$

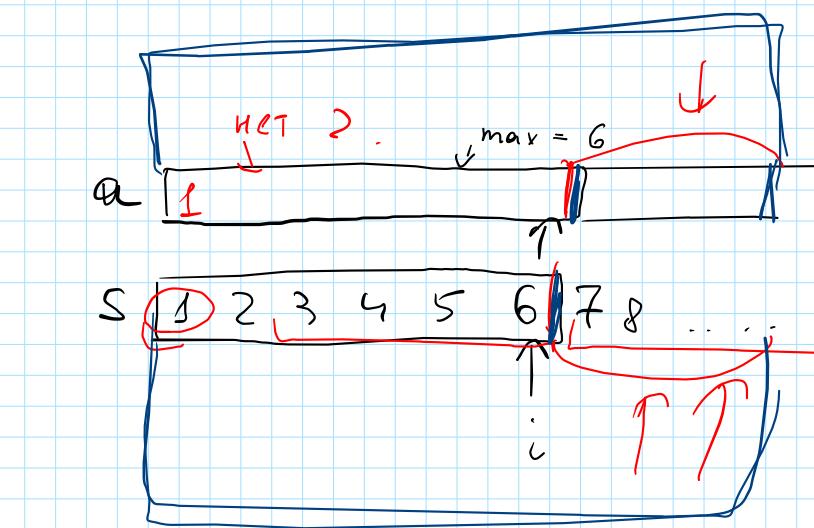


$$\text{Set}(a[1 \dots i]) = \text{Set}(s[1 \dots i]) \rightarrow \text{per } no$$

$$d[2] = 1$$

$$d[\underline{\underline{1}}] = -1$$

$$d[2] = (\text{kon-ro} 2ba) - (\text{kon-ro} 2bs)$$



$$\max(a[1 \dots i]) = \max(s[1 \dots i])$$

2. Дан массив $[a_1, \dots, a_n]$ неотрицательных целых чисел и число K . Отрезок $[a_l, \dots, a_r]$ назовём интересным, если в нём не более одного нулевого элемента, а сумма всех элементов не превосходит числа K . Найдите число интересных подотрезков массива.

0
0 3
0 3

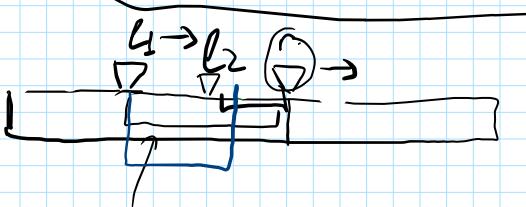
$$a = \left[\begin{array}{ccc} 1 & 0 & 3 \\ 1 & 0 & 4 \end{array} \right] \quad K = 4.$$

The diagram illustrates the divide-and-conquer algorithm for finding the k -th smallest element in an array. It shows an array $[l, r]$ with indices l and r . A pivot element p is selected from the array. The array is partitioned into three regions: elements less than p , elements equal to p , and elements greater than p . The region of elements less than p is highlighted in yellow. The size of this yellow region is labeled as $\frac{n(n-1)}{2}$. The total number of elements in the array is n .

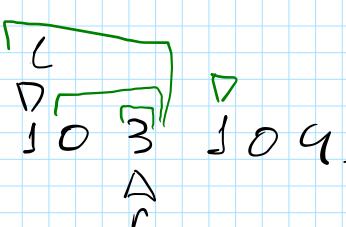
нога - к [i, ɪ] - унт, то.

$\forall [l, r] \in \text{UNT}$

$$i \leq l \leq r \leq j \quad l = 1.$$



WIT



$$K=4$$

$$S = 4$$

$$ans = 1 + 2 + \boxed{3}$$

$$= 1 + 2 + \underline{2}$$