

Домашнее задание 1

Сдать задание нужно до 26 октября. (16:00).

Контеcт: <https://contest.yandex.ru/contest/14656/enter/>

Контеcт для заочников: <https://contest.yandex.ru/contest/14907/enter/>

От каждой задачи нужно решить только один свой вариант. Варианты прописаны в ведомости:

https://drive.google.com/open?id=1RqX6nsJVEaEKUrAOnhR847Z9rco44GT8Z_fJbUu1RP0

Сдаваемый код должен удовлетворять минимальным правилам кодирования:

<https://docs.google.com/document/d/1NxQAhyhoRD59sURVDn1eUniqDWQKxqergrJgx-bRFo>

Задача № 1 (5 баллов)

Во всех задачах из следующего списка следует написать структуру данных, обрабатывающую команды push* и pop*.

Формат входных данных.

В первой строке количество команд n. $n \leq 1000000$.

Каждая команда задаётся как 2 целых числа: a b.

a = 1 - push front

a = 2 - pop front

a = 3 - push back

a = 4 - pop back

Для очереди используются команды 2 и 3. Для дека используются все четыре команды.

Если дана команда pop*, то число b - ожидаемое значение. Если команда pop вызвана для пустой структуры данных, то ожидается "-1".

Формат выходных данных.

Требуется напечатать YES - если все ожидаемые значения совпали. Иначе, если хотя бы одно ожидание не оправдалось, то напечатать NO.

1_1. Реализовать очередь с динамическим зацикленным буфером.

in	out
3 3 44 3 50 2 44	YES
2 2 -1 3 10	YES
2 3 44 2 66	NO

1_2. Реализовать дек с динамическим зацикленным буфером.

in	out
3 1 44 3 50	YES

2 44	
2 2 -1 1 10	YES
2 3 44 4 66	NO

1_3. Реализовать очередь с помощью двух стеков. Использовать стек, реализованный с помощью динамического буфера.

in	out
3 3 44 3 50 2 44	YES
2 2 -1 3 10	YES
2 3 44 2 66	NO

1_*. Реализовать очередь при помощи нескольких стеков. Каждая операция pop front и push back должна выполняться за $O(1)$.

Задача № 2 (5 баллов)

Решение всех задач данного раздела предполагает использование **кучи**.

2_1. Жадина.

Вовочка ест фрукты из бабушкиной корзины. В корзине лежат фрукты разной массы. Вовочка может поднять не более K грамм. Каждый фрукт весит не более K грамм. За раз он выбирает несколько самых тяжелых фруктов, которые может поднять одновременно, откусывает от каждого половину и кладет огрызки обратно в корзину. Если фрукт весит нечетное число грамм, он откусывает большую половину. Фрукт массы 1гр он съедает полностью.

Определить за сколько подходов Вовочка съест все фрукты в корзине.

Формат входных данных. Вначале вводится n - количество фруктов и n строк с массами фруктов.

$n \leq 50000$.

Затем K - "грузоподъемность". $K \leq 1000$.

Формат выходных данных. Неотрицательное число - количество подходов к корзине.

in	out
3 1 2 2 2	4
3 4 3 5 6	5

7 1 1 1 1 1 1 3	3
-----------------------	---

2_2. Быстрое сложение.

Для сложения чисел используется старый компьютер. Время, затрачиваемое на нахождение суммы двух чисел равно их сумме.

Таким образом для нахождения суммы чисел 1,2,3 может потребоваться разное время, в зависимости от порядка вычислений.

$$((1+2)+3) \rightarrow 1+2 + 3+3 = 9$$

$$((1+3)+2) \rightarrow 1+3 + 4+2 = 10$$

$$((2+3)+1) \rightarrow 2+3 + 5+1 = 11$$

Требуется написать программу, которая определяет минимальное время, достаточное для вычисления суммы заданного набора чисел.

Формат входных данных. Вначале вводится n - количество чисел. Затем вводится n строк - значения чисел (значение каждого числа не превосходит 10^9 , сумма всех чисел не превосходит $2 \cdot 10^9$).

Формат выходных данных. Натуральное число - минимальное время.

in	out
5 5 2 3 4 6	45
5 3 7 6 1 9	56

2_3. Тупики.

На вокзале есть некоторое количество тупиков, куда прибывают электрички. Этот вокзал является их конечной станцией. Дано расписание движения электричек, в котором для каждой электрички указано время ее прибытия, а также время отправления в следующий рейс. Электрички в расписании упорядочены по времени прибытия. Когда электричка прибывает, ее ставят в свободный тупик с минимальным номером. При этом если электричка из какого-то тупика отправилась в момент времени X, то электричку, которая прибывает в момент времени X, в этот тупик ставить нельзя, а электричку, прибывающую в момент X+1 — можно.

В данный момент на вокзале достаточное количество тупиков для работы по расписанию.

Напишите программу, которая по данному расписанию определяет, какое минимальное количество тупиков требуется для работы вокзала.

Формат входных данных. Вначале вводится n - количество электричек в расписании. Затем вводится n строк для каждой электрички, в строке - время прибытия и время отправления. Время - натуральное число от 0 до 10^9 . Строки в расписании упорядочены по времени прибытия.

Формат выходных данных. Натуральное число - минимальное количество тупиков.

Максимальное время: 50мс, память: 5Мб.

in	out
1 10 20	1
2 10 20 20 25	2

3 10 20 20 25 21 30	2
------------------------------	---

2_4. Скользящий максимум.

Дан массив целых чисел $A[0..n]$, n не превосходит 10^8 . Также задан размер некоторого окна (последовательно расположенных элементов массива) в этом массиве k , $k \leq n$. Требуется для каждого положения окна (от 0 и до $n-k$) вывести значение максимума в окне. Скорость работы $O(n \log n)$, память $O(n)$.

Формат входных данных. Вначале вводится n - количество элементов массива. Затем вводится n строк со значением каждого элемента. Затем вводится k - размер окна.

Формат выходных данных. Разделенные пробелом значения максимумов для каждого положения окна.

in	out
3 1 2 3 2	2 3
9 0 7 3 8 4 5 10 4 6 4	8 8 8 10 10 10