

Лабораторная 1. Первая половина тем

А. Простая сортировка

2 секунды, 64 мегабайта

В этой задаче вам нужно реализовать любую из пройденных сортировок, работающих за время $O(n \log n)$. Использовать встроенные в язык сортировки и структуры данных запрещается.

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания.

Входные данные

В первой строке содержится число n ($1 \leq n \leq 100\,000$) — количество элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 .

Выходные данные

Выведите этот же массив в порядке неубывания.

входные данные
10 1 8 2 1 4 7 3 2 3 6
выходные данные
1 1 2 2 3 3 4 6 7 8

В. Сортировка подсчетом

1 секунда, 64 мегабайта

А в этой задаче вам нужно реализовать сортировку подсчетом. Использовать другие сортировки запрещается.

Дан массив из n элементов, которые принимают целые значения от 0 до 100. Отсортируйте этот массив в порядке неубывания элементов.

Входные данные

В первой строке содержится число n ($1 \leq n \leq 200\,000$) — количество элементов в массиве. Во второй строке находятся n целых чисел, от 0 до 100 каждое.

Выходные данные

Выведите отсортированный массив.

входные данные
5 7 3 4 2 5
выходные данные
2 3 4 5 7

С. Количество инверсий

5 секунд, 256 мегабайт

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \dots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$.

Входные данные

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 500\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A ($0 \leq a_i \leq 10^6$).

Выходные данные

В выходной файл выведите одно число — ответ на задачу.

входные данные
4 1 2 4 5
выходные данные
0

входные данные
4 5 4 2 1
выходные данные
6

Д. Печать вслепую

4 секунды, 512 мегабайт

Это интерактивная задача. Ваше решение должно следовать определенному протоколу взаимодействия (см. секцию «Протокол взаимодействия»).

Интерактором фиксируется клавиатура a с n кнопками. На каждой кнопке клавиатуры написан символ с кодом от 1 до n . Символы на кнопках не повторяются. Таким образом, клавиатуру можно задать перестановкой чисел от 1 до n , например: $a = [4, 5, 1, 2, 7, 3, 6]$.

Ваша цель — восстановить клавиатуру, то есть определить какой символ написан на каждой кнопке. Для этого вы можете попросить интерактор нажать на любую кнопку, чтобы напечатать соответствующий символ. В ответ интерактор сообщает вам, правда ли, что напечатанный символ меньше предыдущего. После печати первого символа интерактор всегда отвечает «нет».

Для рассмотренной выше клавиатуры $a = [4, 5, 1, 2, 7, 3, 6]$, ответы на последовательность запросов 1, 4, 3, 6 будут «нет», «да», «да», «нет», так как $a_1 = 4$ печатается первой, $a_4 = 2 < 4 = a_1$, $a_3 = 1 < 2 = a_1$ и не верно, что $a_6 = 3 < 1 = a_1$.

Протокол взаимодействия

Интерактор в первой строке сообщает вашей программе целые числа n ($1 \leq n \leq 10^4$) и q ($1 \leq q \leq 10^6$) — размер загаданной перестановки и ограничение на количество запросов.

После этого взаимодействие вашей программы с интерактором должно следовать следующему протоколу:

- Ваша программа направляет интерактору команду «+ x », где x — целое число от 1 до n , означающую, что вы нажимаете на кнопку номер x ;
- Интерактор сообщает вам результат нажатия на кнопку x : если символ, соответствующий этой кнопке, меньше предыдущего напечатанного, интерактор пишет на новой строке «YES» (без кавычек), иначе «NO».

Каждый ответ интерактора завершается переводом строки. Каждый запрос вашей программы должен завершаться переводом строки и сброс буфера потока вывода (`cout.flush()` в C++, `sys.stdout.flush()` в Python).

Первый ответ интерактора в момент, когда еще ни один символ не напечатан — всегда «NO».

Когда ваша программа может восстановить расположение символов на клавиатуре, вместо запроса «+ x» следует вывести «! a1 ... an», где a_i — код символа на i -й кнопке клавиатуры.

Если ваш ответ верен, интерактор выведет «OK», иначе — «FAIL», после чего завершится. Считав последний ответ интерактора, ваша программа должна также завершиться. Если ваша программа не завершается, выдается вердикт TL (Time Limit Exceeded).

Если ваша программа делает больше q запросов вида «+ x» к интерактору, интерактор выводит «FAIL» и завершается. В таком случае вашему решению выставляется вердикт «WA» (Wrong Answer).

Система оценки

Каждый тест оценивается от 1 до 5 баллов, в зависимости от сложности теста.

Тесты состоят из пяти групп с различными ограничениями на разрешенное количество запросов к интерактору. В последних двух группах $q \approx 40n$ и $q \approx 20n$ соответственно. Тестирование на очередной группе будет выполняться только при условии прохождения всех предыдущих групп, кроме группы номер 2, баллы за которую могут быть получены даже если группа 1 не пройдена.

По каждой группе тестов дается полная информация о проверке.

входные данные
4 30
NO
NO
YES
YES
YES
OK
выходные данные
+ 1
+ 2
+ 3
+ 4
+ 1
! 1 4 3 2

В тесте из условия интерактор сообщает, что на клавиатуре 4 кнопки, и у вас есть 30 запросов, чтобы отгадать все символы.

Ответ на второй запрос дает информацию о том, что $a_2 > a_1$, а следующие три — что $a_2 > a_3 > a_4 > a_1$. По этой информации можно однозначно восстановить, что $a_1 = 1, a_4 = 2, a_3 = 3$ и $a_2 = 4$.

Полученный ответ выводится, принимается интерактором как правильный, после чего обе программы завершаются.

Е. Хип ли?

1 секунда, 256 мегабайт

Задачи - Codeforces

Структуру данных Heap можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого $1 \leq i \leq n$ выполняются следующие условия:

- Если $2i \leq n$, то $a[i] \leq a[2i]$
- Если $2i + 1 \leq n$, то $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он Heap'ом.

Входные данные

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел по модулю не превосходящих $2 \cdot 10^9$.

Выходные данные

Выведите «YES», если массив является Heap'ом и «NO» в противном случае.

входные данные
5
1 0 1 2 0
выходные данные
NO

входные данные
5
1 3 2 5 4
выходные данные
YES

Ф. Хипуй!

3 секунды, 256 мегабайт

В этой задаче вам необходимо организовать структуру данных Heap для хранения целых чисел, над которой определены следующие операции:

- Insert (X) — добавить в Heap число X;
- Extract — достать из Heap наибольшее число (удалив его при этом).

Эту задачу нужно решить без использования встроенных структур данных для поиска максимального числа.

Входные данные

Во входном файле записано количество команд n ($1 \leq n \leq 100\,000$), потом последовательность из n команд, каждая в своей строке.

Каждая команда имеет такой формат: "0 <число>" или "1", что означает соответственно операции Insert (<число>) и Extract. Добавляемые числа находятся в интервале от 1 до 10^7 включительно.

Гарантируется, что при выполнении команды Extract в структуре находится по крайней мере один элемент.

Выходные данные

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды Extract.

входные данные
7 0 100 0 10 1 0 5 0 30 0 50 1
выходные данные
100 50

G. Префиксные суммы

2 секунды🕒, 256 мегабайт

Дан массив целых чисел. Необходимо для каждого из m запросов выводить сумму на отрезке $[l, r]$.

Входные данные

В первой строке заданы числа n и m — размер массива и количество запросов ($1 \leq n, m \leq 10^6$). Во второй строке n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — элементы массива a . Далее следуют m строк вида $l_i r_i$, для которых нужно посчитать сумму на отрезке $[l_i, r_i]$.

Выходные данные

Для каждого запроса выведите на отдельной строке сумму на отрезке $[l_i, r_i]$.

входные данные
6 4 1 4 3 7 10 8 1 3 2 5 3 6 1 6
выходные данные
8 24 28 33

H. Скобки

2 секунды🕒, 256 мегабайт

Требуется определить, является ли правильной данная последовательность круглых, квадратных и фигурных скобок.

Входные данные

В единственной строке входного файла записано подряд N скобок ($1 \leq N \leq 10^5$).

Выходные данные

В выходной файл вывести «YES», если данная последовательность является правильной, и «NO» в противном случае.

входные данные
()
выходные данные
YES

входные данные
{[]}{}

выходные данные
YES

входные данные
[]([])
выходные данные
NO

Скобочная последовательность называется правильной, если ее можно получить из какого-либо математического выражения вычеркиванием всех символов, кроме скобок.

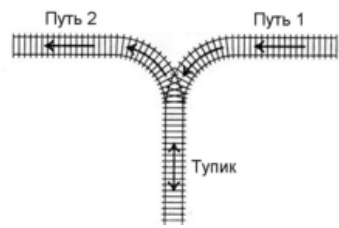
Формальное определение правильной скобочной последовательности таково:

- Пустая последовательность является правильной.
- Если A — правильная скобочная последовательность, то (A) , $[A]$ и $\{A\}$ — правильные скобочные последовательности.
- Если A и B — правильные скобочные последовательности, то AB — правильная скобочная последовательность.

I. Сортировка вагонов

2 секунды🕒, 256 мегабайт

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.



Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика).

Входные данные

Вводится число N — количество вагонов в поезде ($1 \leq N \leq 2000$). Далее идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N , каждое из которых встречается ровно один раз.

Выходные данные

Если сделать так, чтобы вагоны шли в порядке от 1 до N , считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите действия, которые нужно проделать с поездом. Каждое действие описывается двумя числами: типом и количеством вагонов:

- если нужно завезти с пути 1 в тупик K вагонов, должно быть выведено сначала число 1, а затем — число K ($K \geq 1$),

- если нужно вывезти из тупика на путь 2 K вагонов, должно быть выведено сначала число 2, а затем — число $K(K \geq 1)$.

Если возможно несколько последовательностей действий, приводящих к нужному результату, выведите любую из них. Если выстроить вагоны по порядку невозможно, выведите одно число 0.

входные данные
3 3 2 1
выходные данные
1 1 1 1 1 1 2 1 2 1 2 1

входные данные
4 4 1 3 2
выходные данные
1 1 1 1 2 1 1 1 1 1 2 1 2 1 2 1

входные данные
3 2 3 1
выходные данные
0

J. Обход файлов

1 секунда, 256 мегабайт

Сережа уже взрослый и вместо игр он гуляет по директориям на своем компьютере. Например, сегодня он посетил `/Documents/programming/cheats/` и `/Documents/programming/2020/solutions/K.txt`.

Поскольку в последнее время он делает это довольно часто, ему хочется иметь удобное для просмотра представление директорий, чтобы можно было видеть, какие директории в какие вложены. Для этого он просит вашей помощи.

Вам требуется по списку директорий вывести их перечисление в алфавитном порядке, где каждая вложенная директория выведена с отступом на два пробела больше, чем ее родительская директория.

Входные данные

В первой строке дано число n — количество директорий на компьютере Сережи ($1 \leq n \leq 10^5$).

В слудующих n строках по одному в строке заданы абсолютные пути ко всем директориям, каждый абсоютный путь — это последовательность вложенных папок, начиная с корневой, разделенная символами `</>`.

Задачи - Codeforces

Гарантируется, что первая директория во всех путях одинаковая и имеет непустое имя. Имена всех директорий состоят из маленьких латинских букв и имеют длину не более 10. Гарантируется, что если директория выведена, то выведены и все, в которые она вложена.

Выходные данные

Выведите перечисление всех директорий, в котором все директории внутри одной упорядочены по алфавиту, вложенные идут сразу после родительской и имеют отступ на два пробела больше, чем у нее.

входные данные
6 root/a root/a/b root/c/x root/a/b/c root root/c
выходные данные
root a b c c x

входные данные
4 a/b/c/d a/b a/b/c a
выходные данные
a b c d

K. Минимум в очереди

4.0 c, 256 мегабайт

Изначально очередь пуста. Приходят запросы

- `+` `x` добавить в конец очереди элемент x
- `-` удалить из начала очереди элемент

Гарантируется, что вторая операция не пытается удалить элемент из пустой очереди. После каждой операции нужно выводить минимум в очереди после него.

Ограничение по времени выставлено так, чтобы решение с `std::set` не проходило.

Входные данные

В первой строке записано единственное число q ($1 \leq q \leq 10^6$) — количество запросов. В следующих q строках записаны сами запросы в описанном выше формате. Все числа во входном файле целые, положительные и не превышают 10^9 .

Выходные данные

Для каждого запроса выведите единственное число — минимум в очереди после выполнения этого запроса. Если после запроса очередь пуста, выводите -1.

В первой строке входных данных содержатся числа n и k ($0 < n, k \leq 10^5$). Во второй строке задаются n чисел первого массива, отсортированного по неубыванию, а в третьей строке — k чисел второго массива. Каждое число в обоих массивах по модулю не превосходит $2 \cdot 10^9$.

Выходные данные

Для каждого из k чисел выведите в отдельную строку число из первого массива, наиболее близкое к данному. Если таких несколько, выведите меньшее из них.

входные данные
5 5 1 3 5 7 9 2 4 8 1 6
выходные данные
1 3 7 1 5

входные данные
10 + 1 + 2 + 3 + 4 + 5 - - - - -
выходные данные
1 1 1 1 1 2 3 4 5 -1

L. Быстрый поиск в массиве

1 секунда🕒, 512 мегабайт

Дан массив из n целых чисел. Все числа от -10^9 до 10^9 .
Нужно уметь отвечать на запросы вида «Сколько чисел имеют значения от l до r »?

Входные данные

Число n ($1 \leq n \leq 10^5$). Далее n целых чисел.

Затем число запросов k ($1 \leq k \leq 10^5$).

Далее k пар чисел l, r ($-10^9 \leq l \leq r \leq 10^9$) — собственно запросы.

Выходные данные

Выведите k чисел — ответы на запросы.

входные данные
5 10 1 10 3 4 4 1 10 2 9 3 4 2 2
выходные данные
5 2 2 0

M. Приближенный двоичный поиск

2 секунды🕒, 256 мегабайт

Даны два массива. Первый массив отсортирован по неубыванию, второй массив содержит запросы — целые числа.

Для каждого запроса выведите число из первого массива наиболее близкое (то есть с минимальным модулем разности) к числу в этом запросе. Если таких несколько, выведите меньшее из них.

Входные данные

N. Очень Легкая Задача

2 секунды🕒, 256 мегабайт

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре, и теперь ему нужно до начала олимпиады успеть сделать еще n копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за x секунд, а другой — за y . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

Входные данные

На вход программы поступают три натуральных числа n, x и y , разделенные пробелом ($1 \leq n \leq 2 \cdot 10^8, 1 \leq x, y \leq 10$).

Выходные данные

Выведите одно число — минимальное время в секундах, необходимое для получения n копий.

входные данные
4 1 1
выходные данные
3

входные данные
5 1 2
выходные данные
4

O. Квадратный корень и квадратный квадрат

2 секунды🕒, 256 мегабайт

Найдите такое число x , что $x^2 + \sqrt{x} = C$, с точностью не менее 6 знаков после точки.

Входные данные

В единственной строке содержится вещественное число $1.0 \leq C \leq 10^{10}$.

Выходные данные
Выведите одно число — искомый x .

входные данные
2.0000000000
выходные данные
1.0

входные данные
18.0000000000
выходные данные
4.0

P. K-best

2 секунды🕒, 256 мегабайт

У Демьяны есть n драгоценностей. Каждая из драгоценностей имеет ценность v_i и вес w_i . С тех пор, как её мужа Джонни уволили в связи с последним финансовым кризисом, Демьяна решила продать несколько драгоценностей. Для себя она решила оставить лишь k лучших. Лучших в смысле максимизации достаточно специфического выражения: пусть она оставила для себя драгоценности номер i_1, i_2, \dots, i_k , тогда максимальной должна быть величина

$$\frac{\sum_{j=1}^k v_{i_j}}{\sum_{j=1}^k w_{i_j}}$$

Помогите Демьяне выбрать k драгоценностей требуемым образом.

Входные данные
На первой строке n и k ($1 \leq k \leq n \leq 100\,000$).

Следующие n строк содержат пары целых чисел v_i, w_i ($0 \leq v_i \leq 10^6, 1 \leq w_i \leq 10^6$, сумма всех v_i не превосходит 10^7 , сумма всех w_i также не превосходит 10^7).

Выходные данные
Выведите k различных чисел от 1 до n — номера драгоценностей. Драгоценности нумеруются в том порядке, в котором перечислены во входных данных. Если есть несколько оптимальных ответов, выведите любой.

входные данные
3 2 1 1 1 2 1 3

выходные данные
1 2

Q. Разделение массива

1 секунда🕒, 256 мегабайт

Дан массив из n положительных целых чисел. Нужно разбить его на k отрезков так, чтобы максимальная сумма на отрезке была минимально возможной.

Входные данные
Первая строка содержит целые числа n и k ($1 \leq k \leq n \leq 10^5$). Вторая строка содержит элементы массива a_i ($1 \leq a_i \leq 10^9$).

Выходные данные
Выведите одно число — минимально возможную максимальную сумму на отрезке.

входные данные
10 4 1 3 2 4 10 8 4 2 5 3
выходные данные
12

R. К-я сумма

2.0 с🕒, 256 мегабайт

Есть два массива a и b , каждый из которых состоит из n чисел. Для каждой пары чисел $(i, j) : 1 \leq i, j \leq n$ выпишем сумму чисел $a_i + b_j$. Найдите в полученном множестве сумм k -ю по возрастанию.

Входные данные
Первая строка содержит целые числа n и k ($1 \leq n \leq 10^5, 1 \leq k \leq n^2$). Вторая строка содержит элементы массива a , третья строка содержит элементы массива b . Все элементы массивов — целые положительные числа, не больше 10^9 .

Выходные данные
Выведите одно число — искомая k -я сумма.

входные данные
5 10 4 2 6 4 8 7 3 1 9 5
выходные данные
9