Деревья поиска

A. Следующий

1 секунда, 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с котором разрешается производить следующие операции:

1. `add(i)` – добавить в множество S число i (если он там уже есть, то множество не меняется)
2. `next(i)` – вывести минимальный элемент множества, не меньший i. Если искомый элемент в структуре отсутствует, необходимо вывести -1.

Входные данные

Исходно множество S пусто. Первая строка входного файла содержит n – количество операций $(1 \le n \le 300\,000)$. Следующие n строк содержат операции. Каждая операция имеет вид либо «`+ i`», либо «`? i`». Операция «`? i`» задает запрос `next(i)`.

Если операция «`+ i`» идет во входном файле в начале или после другой операции «+», то она задает операцию `add(i)`. Если же она идет после запроса «?», и результат этого запроса был y, то выполняется операция add((i+y)mod 10^9).

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9.

Выходные данные

Для каждого запроса выведите одно число – ответ на запрос.

входные данные
6 + 1 + 3 + 3 ? 2 + 1 ? 4
выходные данные
3 4

В этой задаче запрещено использовать STL.

B. Река

3 секунды, 256 мегабайт

Во Флатландии протекает богатая рыбой река Большой Флат. Много лет назад река была поделена между n рыболовными предприятиями, каждое из которых получило непрерывный отрезок реки. При этом i-е предприятие, если рассматривать их по порядку, начиная от истока, изначально получило отрезок реки длиной a_i.

С тех пор с рыболовными предприятиями во Флатландии k раз происходили различные события. Каждое из событий было одного из двух типов: банкротство некоторого предприятия или разделение некоторого предприятия на два. При некоторых событиях отрезок реки, принадлежащий предприятию, с которым это событие происходит, делится на две части. Каждый такой отрезок имеет длину большую или равную 2. Деление происходит по следующему правилу. Если отрезок имеет четную длину, то он делится на две равные части. Иначе он делится на две части, длины которых различаются ровно на единицу, при этом часть, которая ближе к истоку реки, имеет меньшую длину.

При банкротстве предприятия происходит следующее. Отрезок реки, принадлежавший обанкротившемуся предприятию, переходит к его соседям. Если у обанкротившегося предприятия один сосед, то этому соседу целиком передается отрезок реки обанкротившегося предприятия. Если же соседей двое, то отрезок реки делится на две части описанным выше способом, после чего каждый из соседей присоединяет к своему отрезку ближайшую к нему часть. При разделении предприятия отрезок реки, принадлежавший разделяемому предприятию, всегда делится на две части описанным выше способом. Разделившееся предприятие ликвидируется, и образуются два новых предприятия. Таким образом, после каждого события каждое предприятие владеет некоторым отрезком реки.

Министерство финансов Флатландии предлагает ввести налог на рыболовные предприятия, пропорциональный квадрату длины отрезка реки, принадлежащего соответствующему предприятию. Чтобы проанализировать, как будет работать этот налог, министр хочет по имеющимся данным узнать, как изменялась величина, равная сумме квадратов длин отрезков реки, принадлежащих предприятиям, после каждого произошедшего события.

Требуется написать программу, которая по заданному начальному разделению реки между предприятиями и списку событий, происходивших с предприятиями, определит, чему равна сумма квадратов длин отрезков реки, принадлежащих предприятиям, в начальный момент времени и после каждого события.

Входные данные

Первая строка входного файла содержит два целых числа: n и p — исходное количество предприятий ($2 \leqslant n \leqslant 100000$) и номер подзадачи ($0 \leqslant p \leqslant 4$) (считайте его просто так).

Вторая строка входного файла содержит n целых чисел a_1, a_2, ..., a_n — длины исходных отрезков реки.

Третья строка входного файла содержит целое число k — количество событий, происходивших с предприятиями ($1 \leqslant k \leqslant 100000$).

Последующие k строк содержат описания событий, i-я строка содержит два целых числа: e_i и v_i — тип события и номер предприятия, с которым оно произошло. Значение e_i = 1 означает, что предприятие, которое после всех предыдущих событий является v_i-м по порядку, если считать с единицы от истока реки, обанкротилось, а значение e_i = 2 означает, что это предприятие разделилось на два.

Гарантируется, что значение v_i не превышает текущее количество предприятий. Гарантируется, что если отрезок предприятия при банкротстве или разделении требуется поделить на две части, то он имеет длину большую или равную 2. Гарантируется, что если на реке осталось единственное предприятие, оно не банкротится.

Выходные данные
Выходной файл должен содержать $(k + 1)$ целых чисел, по одному в строке. Первая строка должна содержать исходную сумму квадратов длин отрезков реки, а каждая из последующих k строк — сумму квадратов длин отрезков реки после очередного события.

входные данные
4 0 3 5 5 4 5 1 1 2 1 1 3 2 2 1 3
выходные данные
75 105 73 101 83 113

C. И снова сумма...

3 секунды, 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с котором разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $sum(l, r)$ — вывести сумму всех элементов x из S, которые удовлетворяют неравенству $l \le x \le r$.

Входные данные
Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \le n \le 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «`+ i`», либо «`? l r`». Операция «`? l r`» задает запрос $sum(l, r)$.

Если операция «`+ i`» идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y, то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9.

Выходные данные
Для каждого запроса выведите одно число — ответ на запрос.

входные данные
6 + 1 + 3 + 3 ? 2 4 + 1 ? 2 4
выходные данные
3 7

D. K-ый максимум

0.5 секунд, 64 мегабайта

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k-й максимум.

Входные данные
Первая строка входного файла содержит натуральное число n — количество команд ($n \le 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \le 10^9$).

Поддерживаемые команды:

- $+1$ (или просто 1): Добавить элемент с ключом k_i.
- 0: Найти и вывести k_i-й максимум.
- -1: Удалить элемент с ключом k_i.

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i-го максимума, он существует.

Выходные данные
Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i-й максимум.

входные данные
11 +1 5 +1 3 +1 7 0 1 0 2 0 3 -1 5 +1 10 0 1 0 2 0 3
выходные данные
7 5 3 10 7 3

E. Машинки

1 секунда, 256 мегабайт

Петя, которому три года, очень любит играть с машинками. Всего у Пети N различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более K машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиной игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

Входные данные
В первой строке содержаться три числа N, K и P ($1 \le K, N \le 100000$, $1 \le P \le 500000$). В следующих P строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

Выходные данные
Выведите единственное число: минимальное количество операций, которое надо совершить Петиной маме.

входные данные
3 2 7 1 2 3 1 3 1 2
выходные данные
4

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2

F. Множества

0.75 секунд, 256 мегабайт

На вступительном контесте в пилотную группу по программированию Вашему другу предложили реализовать структуру данных для хранения множеств чисел. Так как он специализируется на истории литературы, данную структуру придётся реализовать Вам.

Структура должна хранить $m + 1$ множеств чисел от 0 до n, пронумерованных от 0 до m включительно, при этом одно число может принадлежать сразу нескольким множествам. Изначально все множества пустые.

Вы должны реализовать следующие операции на этой структуре:

1. `ADD e s`
   Добавить в множество №s ($0 \le s \le m$) число e ($0 \le e \le n$).
2. `DELETE e s`
   Удалить из множества №s ($0 \le s \le m$) число e ($0 \le e \le n$). Гарантируется, что до этого число e было помещено в множество
3. `CLEAR s`
   Очистить множество №s ($0 \le s \le m$).
4. `LISTSET s`
   Показать содержимое множества №s ($0 \le s \le m$) в возрастающем порядке, либо -1, если множество пусто.
5. `LISTSETSOF e`
   Показать множества, в которых лежит число e ($0 \le e \le n$), либо -1, если этого числа нет ни в одном множестве.

Входные данные
Сначала вводятся числа N ($1 \le N \le 9223372036854775807$), M ($1 \le M \le 100000$) и K ($0 \le K \le 100000$) — максимальное число, номер максимального множества и количество запросов к структуре данных. Далее следуют K строк указанного формата запросов.

Выходные данные
На каждый запрос `LISTSET` Ваша программа должна вывести числа — содержимое запрошенного множества или -1, если множество пусто.

На каждый запрос `LISTSETSOF` Ваша программа должна вывести числа — номера множеств, содержащих запрошенное число, или -1, если таких множеств не существует.

На прочие запросы не должно быть никакого вывода.

Гарантируется, что правильный вывод программы не превышает одного мегабайта.

входные данные
10 10 9 ADD 1 1 ADD 1 2 ADD 2 1 LISTSET 1 LISTSETSOF 1 DELETE 1 1 LISTSET 1 CLEAR 1 LISTSET 1

выходные данные
1 2 1 2 2 -1

Эту задачу можно (и нужно!) решать, используя std::set и std::map.

G. Гонка с дозарядкой

1 секунда, 256 мегабайт

Есть $Y + 1$ гоночная трасса. i-я трасса – горизонтальный отрезок $(0, i)$ – (X, i). Есть n заправок. j-я заправка представляет собой вертикальный отрезок (x_j, y_{j1}) – (x_j, y_{j2}). Проезжая по i-й трассе, машина начинает в точке $(0, i)$ и двигается прямолинейно равномерно к точке (X, i), тратя на каждую единицу расстояния одну единицу бензина. Если в какой-то момент машина проезжает заправку (точка-машина лежит на отрезке-заправке), то бак машины мгновенно заполняется до максимума. Если в какой-то момент бензин закончился, а машина не находится в точке заправки или точке (X, i), трасса считается не пройденной. Для каждого i от 0 до Y определите, какой минимальный объём бака должна иметь машина, чтобы пройти i-ю трассу. Машина начинает с полным баком.

Входные данные
На первой строке целые числа n, Y, X ($1 \le n, Y, X \le 200\,000$).

Следующие n строк содержат по три целых числа x_i, y_{i1}, y_{i2} – описания заправок ($0 < x_i < X$, $0 \le y_{i1} < y_{i2} \le Y$).

Выходные данные
Выведите $Y + 1$ целое число – ответы для всех трасс.

входные данные
3 5 10 4 1 2 7 2 5 2 4 5

выходные данные
10 6 4 7 5 5

входные данные
1 1 2 1 0 1
выходные данные
1 1

Эту задачу можно решать, используя `std::set`