Mail.Ru Group 796,33
Компания

21 апреля в 16:49

Итоги второго раунда Russian Code Cup 2017

Спортивное программирование*, Алгоритмы*, Блог компании Mail.Ru Group

16 апреля прошёл второй квалификационный раунд Russian Code Cup 2017, на котором были побиты рекорды посещаемости за последние тр года. По традиции чуть-чуть хвастаемся результатами и выкладываем разбор задач.

A. Очень важные гости
B. Наименьшее общее кратное
C. Портим порядок
D. Красно-чёрное дерево
E. Изучение массива

В этот раз зарегистрировалось 5262 программиста (из них 647 новых участников). В отборочный раунд мы взяли по результатам ещё 201 участника. Все пять задач успешно решили 10 человек. Первые три места ребята заняли с очень большим отрывом от остальных:

1. На первом месте Andrei Popa из Румынии.
2. Второе место занял Lam Nguyen, отстав всего на две минуты штрафного времени.
3. На третьем месте с небольшим отрывом в пять минут оказался Владислав Епифанов из Нижнего Новгорода, Россия.

Кроме того, в топ-10 попали:

- Адам Бардашевич, Мозырь, Беларусь
- Mahmoudian Arash, Рейна, Иран
- Олег Давыдов, Санкт-Петербург, Россия
- George Wu, Китай
- Юрий Писарчик, Минск, Беларусь
- Роман Удовиченко, Столбцы, Беларусь
- Nguyen Trung, Вьетнам

Мы поздравляем всех прошедших в отборочный раунд. Остальные же могут подготовиться к третьему квалификационному раунду, он пройд субботу, 29 апреля, с 14:00 до 16:00 по московскому времени. Ну а теперь — разбор.

A. Очень важные гости

Ограничение по времени — 1 секунда
Ограничение по памяти — 256 мегабайт

На открытие нового кампуса Университета города N планируют прибыть nm очень важных гостей. Церемония будет проходить в зале, котор имеет форму прямоугольника, места в зале организованы в n рядов по m мест, ряды пронумерованы от 1 до n, места в каждом ряду от 1 до место в i-м ряду обозначается как (i, j).

Организаторы церемонии открытия пронумеровали гостей от 1 до nm в соответствии с их важностью, чем больше — тем важнее. Самый важ гость — мэр города — имеет номер nm. Известно, что мэр планирует сесть на место $(1, 1)$. Теперь необходимо рассадить остальных гостей. П этом гостей необходимо расположить так, чтобы гость с большим номером находился не дальше от мэра, чем гость с меньшим номером. Расстоянием между двумя местами (r_1, s_1) и (r_2, s_2) считается значение $|r_1 - r_2| + |s_1 - s_2|$.

Помогите организаторам церемонии рассадить гостей.

Формат входных данных

Входные данные содержат несколько тестовых наборов. В первой строке задано количество тестов t ($1 \le t \le 400$).

Каждый из тестов описывается двумя целыми числами: n и m ($1 \le n, m \le 20$).

Формат выходных данных

Для каждого теста выведите, как будет выглядеть зал после размещения там гостей.

Выведите n строк, в каждой по m чисел, j-е число i-й строки должно быть равно важности гостя, который будет сидеть на месте (i, j).

Если подходящих способов рассадить гостей несколько, выведите любой из них.

Примеры

Входные данные
```
2
2 3
3 2
```

Выходные данные
```
6 4 2
5 3 1
6 4
5 2
3 1
```

Разбор задачи

Есть два способа решить эту задачу. Первый заключается в том, чтобы рассаживать гостей по диагоналям, начиная от позиции $(1, 1)$. Треб некоторая аккуратность в реализации, чтобы верно учесть случаи $n < m$ и $m < n$. Второй способ позволяет не задумываться о том, в какую сторону вытянут прямоугольник. Просто запустим обход в ширину от места $(1, 1)$ и будем рассаживать гостей, начиная от максимального номера, в порядке извлечения мест из очереди.

B. Наименьшее общее кратное

Ограничение по времени — 2 секунды
Ограничение по памяти — 256 мегабайт

Хорошо известно понятие наименьшего общего кратного — минимального числа, которое делится на каждое из заданных. Это понятие можн обобщить на другие математические понятия. Например, на обыкновенные дроби.

Даны две дроби. Требуется найти их наименьшее общее кратное — такую наименьшую положительную несократимую дробь p / q, что при её делении на каждую из данных дробей в частном получается целое число.

Формат входных данных

Входные данные содержат несколько тестовых примеров. В первой строке входных данных задано число t — количество тестов ($1 \le t \le 50\,00$

Каждый тест описывается одной строкой, которая содержит четыре числа a, b, c, d, которые задают дроби a / b и c / d ($1 \le a, b, c, d \le 10^9$). Гарантируется, что a / b и c / d — несократимые дроби.

Формат выходных данных

Для каждого теста выведите на отдельной строке наименьшее общее кратное дробей a / b и c / d — числитель и знаменатель искомой несокр дроби через пробел.

Примеры

Входные данные
2
9 5 12 5
1 10 3 100

Выходные данные
36 5
3 10

Разбор задачи

Пусть p/q нацело делится на a/b и c/d, при этом все дроби несократимые. Тогда целыми числами являются $(p/q):(a/b) = (p \bullet b)/(q \bullet a)$ и $(p/q):(c/d) = (p \bullet d)/(q \bullet c)$.

$p \bullet b$ делится на $q \bullet a$. Поскольку b и a взаимно просты, p делится на a. p и q тоже взаимно просты, поэтому b делится на q. Аналогично p делится на c, d делится на q.

Таким образом, p делится $lcm(a, c)$, q делит $gcd(b, d)$. Дробь $lcm(a, c)/gcd(b, d)$ является наименьшей подходящей дробью, и она делится на a/b и c/d. Таким образом, ответ равен $lcm(a, c)/gcd(b, d)$.

C. Портим порядок

Ограничение по времени — 2 секунды
Ограничение по памяти — 256 мегабайт

На ковре в комнате у Димы в ряд нарисовано n точек. По удивительному совпадению у него также есть n кубиков, весом $1, 2, ..., n$ грамм, соответственно. Дима наигрался с кубиками и выложил их в ряд, по одному на точку. Теперь он собирался переложить их так, чтобы они шли слева направо по возрастанию веса, но немного отвлёкся. В этот момент в комнату вошёл вредный мальчик Вадим, и решил напакостить Диме.

Вадим знает, что Дима ещё маленький, и будет упорядочивать кубики по возрастанию веса следующим образом. Он будет каждый раз искать самый лёгкий кубик, который ещё не стоит на своём месте, и менять его с тем, который стоит там вместо него.

Вадим — вредный мальчик, поэтому он хочет напакостить побольше. Он уже вытащил из ряда несколько кубиков и теперь хочет их расставить обратно таким способом, чтобы Дима совершил максимальное число обменов. После возвращения кубиков те кубики, которые Вадим не вынул, должны остаться на своих местах, около каждой точки должно оказаться ровно по одному кубику.

Как именно ему следует расположить кубики?

Формат входных данных

Входные данные содержат несколько тестовых наборов. В первой строке задано количество тестов t.

Каждый тест описывается следующим образом.

В первой строке описания теста содержится число n — общее количество кубиков ($1 \le n \le 10^5$). В следующей строке содержится n чисел a_i ($0 \le n$)

Если a_i равно 0, то на i-м месте пока ничего не стоит, этот кубик Вадим вынул. Иначе a_i равно массе кубика, который стоит на i-м месте.

Гарантируется, что массы присутствующих кубиков различны. Вадим должен вернуть в ряд на пустые места в точности те кубики, которых т нет.

Сумма n по всем тестам одних входных данных не превосходит 10^5.

Формат выходных данных

Для каждого теста выведите две строки.

В первой строке выведите количество обменов, которые придётся совершить Диме для сортировки кубиков.

Во второй строке выведите n чисел — массы кубиков в порядке слева направо, которые получатся при оптимальной расстановке кубиков Ва Обратите внимание, что кубики, которые Вадим не вынул сразу, должны остаться на своих местах.

Если существует несколько возможных решений, которые приводят к максимальному числу обменов, выведите любое из них.

Примеры

Входные данные
3

```
2
0 0
4
4 0 0 3
5
0 4 0 2 5
```

Выходные данные

```
1
2 1
3
4 1 2 3
2
3 4 1 2 5
```

Разбор задачи

В задаче требуется дополнить исходный массив до перестановки чисел от 1 до n таким образом, чтобы сортировка выбором сделала наибо число обменов.

Для начала допустим, что мы уже сгенерировали какую-то перестановку и хотим узнать, сколько обменов совершит сортировка выбором. Д этого будем рассматривать перестановку как набор циклов. Рассмотрим цикл, в котором содержится минимальное число, стоящее не на сво месте. Длина этого цикла больше чем 1. Нетрудно заметить, что при одном обмене длина соответствующего цикла уменьшается на один и появляется новый цикл длины 1. Теперь заметим, что цикл длины L уменьшит свою длину ровно $L - 1$ раз в процессе сортировки, из чего следует, что суммарно длины циклов уменьшатся ровно на $n - c$, где c — количество циклов.

Таким образом, нам нужно дополнить массив до перестановки так, чтобы количество циклов получилось минимальным. Для этого рассмотр перестановку как ориентированный граф, где если $a[i] = j$, то в графе есть ребро из вершины i в вершину j. Граф разбился на циклы и цепо Все циклы, которые есть в этом графе, сохранятся и после добавления недостающих элементов, а все имеющиеся цепочки (вершину, не имеющую ни входящих, ни исходящих рёбер, будем считать цепочкой длины 0) можно объединить в один большой цикл. Пройдёмся по все имеющимся цепочкам и будем добавлять ребро из конца одной цепочки в начало другой. Когда все цепочки соединятся в одну, добавим ре из её конца в начало, и она станет циклом.

D. Красно-чёрное дерево

Ограничение по времени — 3 секунды
Ограничение по памяти — 256 мегабайт

Знали ли вы, что в большинстве стандартных библиотек структура данных «множество» реализовано с использованием красно-черного дере этой задаче вам необходимо найти количество способов покрасить заданное двоичное дерево так, чтобы оно стало красно-черным. Ответ необходимо вывести по модулю $10^9 + 7$.

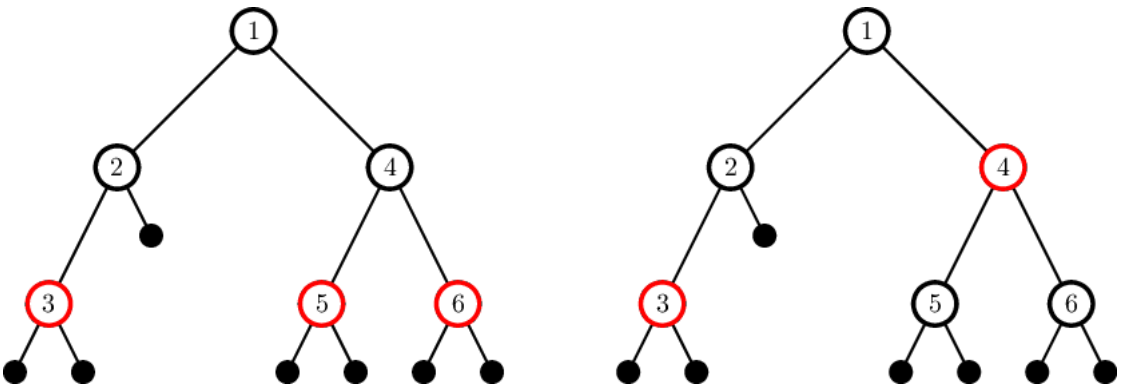
Рассмотрим двоичное дерево. Если у вершины меньше двух детей, добавим на место недостающих детей фиктивные вершины. Дерево назыв красно-черным, если выполняются следующие свойства:

1. Каждая вершина покрашена в один из двух цветов: красный или черный.
2. Корень дерева и все добавленные фиктивные вершины покрашены в черный цвет.
3. Родитель вершины, покрашенной в красный цвет, покрашен в черный цвет.
4. Все пути от корня до добавленных фиктивных листьев содержат одинаковое количество черных вершин.

Обратите внимание, что родитель вершины, покрашенной в черный цвет, может быть покрашен в черный цвет.

Две раскраски дерева называются различными, если существует вершина, которая покрашена в разные цвета в этих раскрасках.

На рисунке приведены два способа покраски дерева из второго теста.



Формат входных данных

В первой строке содержится одно целое число n — количество вершин в дереве ($1 \le n \le 500\,000$).

Следующие n строк описывают дерево. В i-й строке содержатся два целых числа l_i и r_i — индексы вершин, являющихся левым и правым дет
либо 0, если соответствующий ребенок отсутствует ($l_i = 0$ или $i < l_i \le n$; $r_i = 0$ или $i < r_i \le n$). Корень дерева имеет номер 1. Гарантируется, что
вводе описано корректное дерево.

Формат выходных данных

Выведите одно целое число — число способов покрасить заданное дерево, чтобы оно стало красно-черным. Ответ необходимо вывести по м
$10^9 + 7$.

Примеры

Входные данные
```
3
2 3
0 0
0 0
```

Выходные данные
```
2
```

Входные данные
```
6
2 4
3 0
0 0
5 6
0 0
0 0
```

Выходные данные
```
2
```

Разбор задачи

Заметив тонкий намёк в первом предложении условия, легко доказать (или проверить в любой книге по структурам данных), что в любом красно-чёрном дереве с n вершинами на пути от корня до листьев $O(logn)$ чёрных вершин. Воспользуемся этим фактом в решении.

Назовём почти красно-чёрным дерево, в котором выполняются все свойства красно-чёрного дерева, но корень дерева покрашен в красный цвет.

С помощью динамического программирования посчитаем для каждого поддерева количество способов покрасить вершины в нём так, чтобы поддерево было красно-чёрным и количество чёрных вершин на пути от корня до листьев равнялось h — или чтобы дерево было почти кра чёрным и количество чёрных вершин на пути от корня до листьев равнялось h.

Сохраним это количество способов в массиве $d[v][h][t]$, где v — номер вершины, являющейся корнем поддерева, h — количество чёрных вершин на пути от корня до листьев, а t обозначает тип покраски: если $t = 0$ — там хранится количество покрасок, чтобы поддерево стало красно-чёрным, а если $t = 1$ — чтобы поддерево стало почти красно-чёрным.

Тогда $d[v][h][0]$ равняется произведению по всем u, являющимся детьми v, значений $(d[u][h-1][0] + d[u][h-1][1])$. А $d[v][h][1]$ равняется произведению по всем u, являющимся детьми v, значений $d[u][h][0]$.

Ответ на задачу будет равен сумме по всем $d[1][h][0]$.

Теперь заметим, что h для всех допустимых раскрасок поддеревьев ограничено некоторой величиной $C = O(log(n))$, и при всех $h > C$ количе покрасок гарантированно будет равняться нулю.

Значит, необходимо всего $O(nlog(n))$ состояний, из каждого из которых идёт $O(1)$ переходов. Следовательно, решение будет работать за $O(nlog(n))$.

E. Изучение массива

Ограничение по времени — 2 секунды
Ограничение по памяти — 256 мегабайт

Вася обожает массивы. Поэтому родители на день рождения подарили ему массив a, состоящий из чисел 1 и -1. Вася сразу же начал его из

Так как Вася также очень любит нули, он решил брать разные подотрезки $a[l_i, ..., r_i]$ массива a и искать в них длину максимального подотрез суммой 0. Если такого подотрезка нет, он считает ответ равным 0. Вася написал на бумажке q отрезков $[l_i, r_i]$ и теперь хочет найти сумму отв на них.

Приведем ответы на выбранные Васей подотрезки в тестовом примере:

- подотрезок $[1, 5]$: максимальный подотрезок с суммой 0 — $[2, 5]$;
- подотрезок $[1, 3]$: максимальный подотрезок с суммой 0 — $[2, 3]$;
- подотрезок $[2, 4]$: максимальный подотрезок с суммой 0 — $[2, 3]$;
- подотрезок $[3, 4]$: подотрезка с суммой 0 нет;
- подотрезок $[3, 5]$: максимальный подотрезок с суммой 0 — $[4, 5]$.

Итого, суммарная длина всех искомых подотрезков в этом тесте равна $4 + 2 + 2 + 0 + 2 = 10$.

Формат входных данных

Входные данные содержат несколько тестовых наборов. В первой строке задано количество тестов t ($1 \le t \le 1000$).

Каждый из следующих t тестов описывается следующим образом: в первой строке описания теста содержится число n — количество элемент массива ($1 \le n \le 10^5$).

В следующей строке содержится n целых чисел a_i — элементы массива ($a_i = -1$ или $a_i = 1$).

В следующей строке содержится число q — количество подотрезков, выписанных Васей ($1 \le q \le 10^5$).

В каждой из следующих q строк содержится два числа l_i, r_i — левая и правая границы i-го подотрезка соответственно ($1 \le l_i \le r_i \le n$)

Гарантируется, что сумма n во всех тестах одних входных данных не превосходит 10^5, а также сумма q во всех тестах одних входных данны превосходит 10^5.

Формат выходных данных

Для каждого теста выведите ответ на него — сумму ответов для всех q подотрезков.

Примеры

Входные данные
```
1
5
1 -1 1 1 -1
5
1 5
1 3
2 4
3 4
3 5
```

Выходные данные
```
10
```

Разбор задачи

Поскольку ограничения на n и q совпадают, будем использовать n вместо $max(n, q)$. Сделаем несколько наблюдений.

Первое наблюдение: ответ на запрос — максимальная длина отрезка $[L, R]$, такого, что $pref_L - 1 = pref_R$, где $pref_i = a1 + a2 + ... + a_i$ — префи сумма массива a ($pref_0 = 0$). Далее будем считать, что мы работаем с массивом $pref_0, pref_1, ..., pref_n$ и по запросу $[l, r]$ требуется найти максимальный по длине подотрезок $[L, R]$ отрезка $[l - 1, r]$, такой, что $pref_L = pref_R$.

Второе наблюдение: $pref_i$ — довольно маленькие ($-n \le pref_i \le n$).

Будем решать задачу офлайн. Воспользуемся методом, во многом похожим на алгоритм Мо. Разобьём все запросы на группы, где в i-й груп содержатся запросы $[l_i, r_i]$, такие, что $i \bullet K \le l_i < (i + 1) \bullet K$ (K — константа, приблизительно равная $sqrt(n)$). В каждой группе отсортируем запр по правой границе. Теперь решим задачу отдельно для каждой группы запросов.

Рассмотрим i-ю группу $[L_i, R_i]$. Будем идти подряд по запросам слева направо (т. е. по неубыванию правой границы). Также будем поддерж два массива, $mostLeft[p]$ и $mostRight[p]$: текущее самое левое вхождение префиксной суммы p, которое также левее R_i, и текущее самое п вхождение префиксной суммы p. Поддерживая эти два массива, также несложно поддерживать ответ на отрезке $[R_i + 1, r]$, где r — правая граница текущего запроса. Для ответа на запрос $[l, r]$ посчитаем за $O(K)$ ответ на $[l, min(r, R)]$, используя информацию из массива $mostRigh$ сравним это значение с текущим ответом на отрезке $[R + 1, r]$.

Итого на запросы каждой группы в сумме мы тратим $O(K \bullet c_i + n)$ времени, где c_i — количество запросов в группе. Так как групп всего $O(n / K)$ получаем суммарное время работы $O(sum_i = 1... \ n / K(K \bullet c_i + n)) = O(K \bullet sum(c_i) + n2 / K)$. При $K = sqrt(n)$ получаем время работы $O(n \bullet sqrt(n))$.

Все на третий раунд!

Ещё раз напоминаем: третий раунд начнётся 29 апреля с 14:00 по московскому времени. Более пяти тысяч участников уже зарегистрирован Будет с кем посоревноваться!

Присоединяйтесь!

rcc2017, rcc, russian code cup, разбор задач, алгоритмы

Комментарии (5)

erwins22 22 апреля 2017 в 10:36

вопросы вызывает задача Б

«Требуется найти их наименьшее общее кратное — такую наименьшую положительную несократимую дробь p / q, что при её делении на каждую из данных др частном получается целое число.»

рассмотрим более простой вариант a/b: 1/q c/d :1/q

тогда q достаточно выбрать как q=b*d*N где N — бесконечно большое натуральное число.

erwin_shrodinger 25 апреля 2017 в 01:17

Если вы возьмёте две одинаковые дроби, то они сами и будут являться НОК'ами.
Это следует и из выражения *lcm(a, c) / gcd(b, d)*.
А шаманство с бесконечным N — это какая-то подмена понятий, разве нет?

fi11er 25 апреля 2017 в 01:17

Тоже не понял формулировку. Сутя по примеру входных данных, нужно найти наибольший общий делитель.

fi11er 25 апреля 2017 в 13:51

А нет, все нормально, это я перепутал нок и нод. Для дробей a/b и c/d нужно найти найти наименьшую дробь p/q такую, что pb/qa и pd/qc — целые, а не а cq/dp.

sat2707 25 апреля 2017 в 01:21 *(комментарий был изменён)*

Попробуйте здесь
Если правы и участвовали — аппелируйте
Если правы и не участвовали — напишите, пожалуйста, я вставлю disclaimer в статью :)

САМОЕ ЧИТАЕМОЕ Разра

Сейчас | Сутки | Неделя | Месяц

Получил 1.2K звезд на GitHub с ужасной архитектурой. Как?
+22 66,2k 110 39

Реализация псевдо-3D в гоночных играх
+90 26,4k 239 16

Индексы в PostgreSQL — 1
+102 24k 430 57

Выбор MQ для высоконагруженного проекта
+30 19,4k 169 51

Алгоритм Джонкера-Волгенанта + t-SNE = супер-сила
+63 17,6k 200 2

ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ

Построение графиков с двумя независимыми осями в Matlab GT
530 2 2

Valve пытается сделать работу саппорта Steam более прозрачной и эффективной GT
1,4k 1 2

Ко дню связи: история IP-телефонии
2k 16 11

Подключаем геймпад от PS1/PS2 к Raspberry pi GT
1,3k 3 0

Считаем до трёх: три
2,3k 12 16

Аккаунт
Войти
Регистрация

Разделы
Публикации
Хабы
Компании
Пользователи
Песочница

Информация
О сайте
Правила
Помощь
Соглашение
Помощь стартапам

Услуги
Реклама
Тарифы
Контент
Семинары

Приложения
Загрузите в App Store
ДОСТУПНО в Google

Служба поддержки Мобильная версия