

А. Новые земли

1 секунда, 256 мегабайт

Исследователи Байтландии нашли новый континент! Земля оказалась очень плодородной за счёт обильного орошения и благоприятного климата. Поэтому правитель Байтландии хочет расширить свою страну и основать города на новом континенте.

Во время исследований было обнаружено n подходящих мест для основания городов. Правитель хочет основать все n городов. Однако перед строителями Байтландии возникла трудная задача — на континенте довольно много водоёмов и рек, поэтому между городами придётся строить мосты. Будучи главным программистом Байтландии, Вам нужно решить эту задачу.

Для простоты опишем реки следующим образом: Вам будут даны координаты двух точек, тогда река представляет собой отрезок, соединяющий их. Обратите внимание, что реки могут иметь общие точки.

Ваша задача — проложить дороги между городами так, чтобы в каждый город можно было попасть из любого города на континенте, при этом построив как можно меньше мостов. Разрешается строить дороги в виде отрезков, соединяющих города по прямой. Если на пути между городами несколько рек, то придётся строить по мосту на каждую реку, даже если реки пересекаются в одной и той же точке.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит три целых числа n, m, k ($1 \leq n \leq 500, 1 \leq m \leq 100, \frac{m}{2} \leq k \leq 50$) — количество мест, подходящих для основания города, границ рек и самих рек.

Следующие m строк набора тестовых данных содержат по два целых числа x, y ($0 \leq x, y \leq 100$) — координаты, описывающие границы рек.

Следующие k строк набора тестовых данных содержат по два целых числа u, v ($1 \leq u, v \leq m, u \neq v$) — номера двух точек, которые образуют реку.

Следующие n строк набора тестовых данных содержат по два целых числа x, y ($0 \leq x, y \leq 100$) — координаты мест, в которых возможно основание города.

Гарантируется, что основывать город на реке не планируется.

Гарантируется, что сумма n по всем наборам тестовых данных не превосходит 500 ($\sum n \leq 500$).

Гарантируется, что сумма m по всем наборам тестовых данных не превосходит 100 ($\sum m \leq 100$).

Гарантируется, что сумма k по всем наборам тестовых данных не превосходит 50 ($\sum k \leq 50$).

Выходные данные

На каждый набор тестовых данных выведите единственное число — минимальное количество мостов, требуемое для соединения всех поселений.

входные данные
3
4 4 4
2 3
2 5
7 5
5 3
1 2
2 3
3 4
4 1
4 1
4 4
4 7
8 3
4 4 3
2 3
2 5
7 5
5 3
1 2
2 3
4 1
4 1
4 4
4 7
8 3
4 4 2
2 3
2 5
7 5
5 3
1 2
3 4
4 1
4 4
4 7
8 3
выходные данные
2
1
0

В. Нечестная игра

1 секунда, 256 мегабайт

Алиса, Боб и Ева собрались вечером, чтобы поиграть в НИМ. Перед ними n кучек камней, i -я кучка состоит из a_i камней. Ева выступает в роли судьи. Изначально Алиса и Боб играют с n кучками. Затем Ева восстанавливает все кучки a_i и убирает из них одну. Алиса и Боб играют с $n - 1$ кучками камней, Ева восстанавливает $n - 1$ кучку и снова убирает одну из них. Так продолжается до тех пор, пока есть хотя бы одна кучка камней.

Ева знает, что в НИМ Алиса почти всегда побеждает, поэтому она хочет, чтобы Боб выиграл больше всего раз. Определите, какое максимальное количество раз Боб сможет победить Алису, если Ева будет убирать кучки оптимально.

Входные данные

В первой строке записано одно целое число n ($1 \leq n \leq 111$) — количество кучек камней

В следующей строке заданы n чисел a_i ($1 \leq a_i \leq 4$) — размеры кучек.

Выходные данные

В единственной строке выведите целое число — ответ на задачу.

входные данные
3
1 2 3

выходные данные
1

входные данные
4
1 1 2 3
выходные данные
1

Во втором тестовом примере сначала побеждает Алиса, затем Ева убирает кучку размера 1, побеждает Боб. Какие бы кучки дальше не убирала Ева, Алиса всегда будет побеждать Боба.

C. Ремонт дорог

1 секунда, 256 мегабайт

Байтландия состоит из n городов, соединённых $n - 1$ двусторонними дорогами. Город под номером f — Килобайттаун — является столицей Байтландии. Из любого города можно попасть в любой другой город.

Президент Байтландии — Нолик Единичковый — хочет отремонтировать все дороги, но ему нужно определить, в каком порядке ремонтировать дороги. Каждая дорога характеризуется степенью износа дорожного полотна w . Чем больше это число, тем больше изношено покрытие.

Для каждого города u известно количество городов, в которые можно попасть из u , двигаясь в направлении от столицы. Обозначим количество таких городов числом c_u . Дорога из u в v называется *загруженной*, если $c_u \geq \frac{c_v}{2}$. Так можно оценить *степень важности* дороги. Если дорога из u в v загружена и имеет износ полотна w , то её *степень важности* — $2 \cdot w$, иначе w .

По имеющимся замерам состояния дорожного покрытия Вам требуется определить порядок ремонта дорог в Байтландии — начиная с более важных и заканчивая менее важными.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит два целых числа n, f ($1 \leq f \leq n \leq 2 \cdot 10^5$) — количество городов и номер столицы Байтландии.

Следующие $n - 1$ строк набора тестовых данных содержат по три целых числа u, v, w ($1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 100$) — номера городов, соединённых дорогой, и состояние дорожного покрытия.

Гарантируется, что сумма n по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum n \leq 2 \cdot 10^5$).

Выходные данные

На каждый набор тестовых данных выведите $n - 1$ строк, содержащих по два целых числа u и v — номера городов, которые следует ремонтировать, в порядке, описанном в условии.

Если возможных решений несколько — выведите любое. Порядок номеров u, v в строке не важен.

входные данные
3
4 1
1 2 1
2 3 1
2 4 1
5 1
1 2 1
2 3 1
2 4 1
5 3 1
6 1
1 2 1
2 3 1
2 4 1
5 3 1
6 3 1

выходные данные
1 2
2 3
2 4
3 5
2 3
1 2
2 4
2 3
1 2
3 5
3 6
2 4

D. Запросы, запросы и ещё раз запросы

1 секунда, 256 мегабайт

На этот раз никто не просит Вас помогать.

Теперь у Вас есть массив a размера n , нужно разработать структуру данных, способную выполнять два типа запросов:

- Вывести максимальную длину и границы подотрезка, содержащего одинаковые числа среди $a_l, a_{l+1} \dots a_r$. Если отрезков с одинаковой длиной несколько, требуется вывести самый левый.
- Изменить значение элемента a_p на x .

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — размер массива a .

Вторая строка набора тестовых данных содержит n целых чисел a_i ($1 \leq a_i \leq 100$) — элементы массива a .

Третья строка набора тестовых данных содержит целое число q ($1 \leq q \leq 2 \cdot 10^5$) — число запросов к структуре данных.

Следующие q строк набора тестовых данных описывают запросы в виде трёх целых чисел

- «1 l r» ($1 \leq l \leq r \leq n$) — узнать наибольшее число подряд идущих одинаковых чисел на подотрезке $[l, r]$.
- «2 p x» ($1 \leq p \leq n, 1 \leq x \leq 100$) — изменить значение элемента a_p на x .

Гарантируется, что сумма n по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum n \leq 2 \cdot 10^5$).

Гарантируется, что сумма q по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum q \leq 2 \cdot 10^5$).

Выходные данные

На каждый запрос первого типа в наборе тестовых данных выведите значение повторяющихся чисел и границы подотрезка, содержащего эти числа в заданом отрезке.

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

Е. Колонизация Марса

3 секунды, 256 мегабайт

Компания SpaceY решила, что в следующем году будет колонизировать Марс. Перед полётом нужно спроектировать структуру колонии. Никакие две ячейки не соединены двумя и более дорогами. Никакая ячейка не соединена сама с собой дорогой.

Колония состоит из n ячеек, которые связаны m двусторонними дорогами. Каждая ячейка может быть трёх типов:

- Жилая ячейка
- Производственная ячейка
- Ячейка отдыха

Колония характеризуется двумя показателями — уровнем производительности и уровнем счастья космонавтов. Они формируются из того, как в колонии расположены те или иные ячейки. Далее следует описание ячеек и того, как формируются показатели. Ячейки считаются соседними, если их соединяет дорога.

- Каждая ячейка отдыха добавляет 1 к уровню счастья. Жилая ячейка рядом с ячейкой отдыха так же повышает этот уровень на 1. Производственная ячейка рядом с ячейкой отдыха уменьшает уровень счастья на 1. Расположенные рядом ячейки отдыха никакого прироста к уровню счастья не дают.
- Производственная ячейка уменьшает уровень счастья на 1. Если рядом с ней расположена жилая ячейка, то производительность возрастает на 1. Две производственных ячейки рядом никак не влияют на показатели колонии.
- Жилые ячейки сами по себе никак не влияют на показатели колонии. Они изменяют показатели, если расположены рядом с другими типами ячеек.

Ваша задача — определить для описанного графа ячеек колонии максимальную производительность, причём уровень счастья в колонии должен быть не менее k .

Входные данные

В первой строке записаны три целых числа n, m, k ($2 \leq n \leq 13, n - 1 \leq m \leq \frac{n \cdot (n-1)}{2}, -5 \leq k \leq 5$) — количество ячеек и дорог в колонии, необходимый уровень счастья в колонии.

В следующих m строках заданы два числа u и v ($1 \leq u, v \leq n, u \neq v$) — номера ячеек колонии, соединённые дорогой.

Выходные данные

В первой строке выведите целое число — производительность колонии.

Затем выведите три строки, описывающие колонию. Выведите целое число h — количество зданий. Затем выведите h чисел — номера ячеек колонии, в которых расположено соответствующее здание.

Описывайте колонию в следующем порядке: жилые ячейки, производственные ячейки, ячейки отдыха.

Если решений несколько, выведите любое.

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

входные данные
4 4 -2 1 2 2 3 3 4 1 4
выходные данные
4 2 2 4 2 1 3 0

Г. Аллокатор Abanta

2 секунды, 512 мегабайт

Вы разрабатываете новую операционную систему Abanta, которая, предположительно, превзойдёт ныне популярную систему Winedoo.

Настало время реализовывать модуль, отвечающий за распределение памяти системы. Для простоты скажем, что память представляет собой массив из n элементов, каждый элемент представляет собой один байт памяти. Байт может быть занят какой-то программой или свободен. Байт может быть занят только одной программой, но может быть освобождён, а затем отдан под использование другой программе.

Как и в любой другой операционной системе, Ваш модуль должен уметь обрабатывать неправильные запросы и выдавать соответствующие ошибки. Требуется реализовать структуру данных, которая отвечает на следующие запросы:

- Выделить память, начиная с ячейки под номером l и заканчивая ячейкой r . В случае успеха, программа должна вывести «OK, allocated memory». Если хотя бы одна ячейка в этом промежутке уже занята другой программой, требуется вывести сообщение об ошибке «Can't allocate memory».
- Освободить память в ячейках с l по r включительно. В случае успеха, программа должна вывести «OK, freed memory». Если память во всех этих ячейках уже свободна, сообщить об этом «Segment is already free». Если ячейки с l по r включительно заняты несколькими (двумя и более) блоками или блок, занимающий эти ячейки не совпадает с границами запроса, то вывести сообщение об ошибке «Can't free memory».
- Среди ячеек от l по r включительно нужно найти самый длинный непрерывный свободный блок памяти. Обратите внимание, что размер этого блока не должен превышать $r - l + 1$.

Изначально все ячейки памяти свободны.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит целое число n ($2^4 - 1 \leq n \leq 2^{36} - 1$) — число байт в памяти компьютера.

Вторая строка набора тестовых данных содержит целое число q ($1 \leq q \leq 2 \cdot 10^5$) — число запросов к модулю операционной системы.

Следующие q строк набора тестовых данных содержат по три целых числа $type, l, r$ ($1 \leq type \leq 3, 0 \leq l \leq r \leq n$) — тип запроса и границы запроса.

Гарантируется, что сумма q по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum q \leq 2 \cdot 10^5$).

Выходные данные

На каждый запрос в наборе тестовых данных Ваша программа должна вывести соответствующее сообщение или число как описано в условии.

входные данные
1 1024 14 1 0 128 2 1 2 2 1 128 1 63 64 2 127 128 2 0 129 2 256 512 3 0 512 3 0 256 1 256 512 3 0 512 2 0 128 3 0 512 3 0 1024
выходные данные
OK, allocated memory Can't free memory Can't free memory Can't allocate memory Can't free memory Can't free memory Segment is already free 384 128 OK, allocated memory 127 OK, freed memory 256 512

G. Разработка программного продукта

2 секунды, 256 мегабайт

В компания Zandex работают n сотрудников, каждый владеет каким-то количеством навыков на определённом уровне. Для простоты уровень владения будем оценивать целым числом u .

Против страны, где располагается компания ввели санкции, поэтому компании требуется разработать свой продукт, заменяющий иностранную платформу Em-sto-gram.

Для разработки аналога требуется m человек, каждый из которых должен иметь необходимый навык и уровень владения им. Например, Андрей владеет языком C++ на уровне 6. Он может занять вакансии C++ начиная с 1 уровня владения до 6 уровня владения включительно. Но Андрей не может работать на вакансии по языку C#, потому что он его не знает.

Сотрудник не может занимать две вакансии. Никакая вакансия не может быть занята двумя людьми. В проекте могут быть одинаковые требования (например, дважды упомянут язык Java на уровне владения 5), поэтому нужно два человека на две вакансии.

Ваша задача — определить, сможет ли компания одолеть санкции и разработать аналог Em-sto-gram.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит одно целое число n ($1 \leq n \leq 10^3$) — количество сотрудников в компании.

Далее следуют описания сотрудников. Первая строка содержит строку s , состоящую из букв латинского алфавита, и целое число k ($1 \leq |s| \leq 10, 1 \leq k \leq 50$) — имя сотрудника и количество его навыков.

Следующие k строк описывают навыки сотрудника в виде строки t и целого числа u ($1 \leq |t|, u \leq 10$) — названия навыка и уровня владения им, строка t состоит из букв латинского алфавита и символов «+», «#».

Аналогично описываются следующие $n - 1$ сотрудников.

Далее следует целое число m ($1 \leq m \leq 500$) — количество требуемых навыков для выполнения задачи.

Следующие m строк набора тестовых данных содержат строку p и целое число v ($1 \leq |p|, v \leq 10$) — название навыка и требуемые уровень владения им, строка p состоит из букв латинского алфавита и символов «+», «#».

Гарантируется, что сумма n по всем наборам тестовых данных не превосходит 10^3 ($\sum n \leq 10^3$).

Гарантируется, что сумма m по всем наборам тестовых данных не превосходит 500 ($\sum m \leq 500$).

Выходные данные

На каждый набор тестовых данных выведите «YES», если возможно выполнить поставленную задачу. Затем выведите m строк в виде строки p , целого числа v и строки s — названия навыка, требуемого уровня владения им и имени сотрудника, который удовлетворяет требованию. Иначе выведите «NO».

Если возможных решений несколько, выведите любое. Порядок требуемых навыков в выводе не важен.

входные данные
3 4 Andrew 1 C++ 6 George 1 C++ 5 Paul 1 Python 3 Peter 1 Python 4 3 C++ 2 C++ 3 Python 4 1 John 2 C++ 3 C 4 1 Lisp 1 4 Kate 2 C++ 1 Python 2 John 1 C++ 3 Mike 2 Python 6 Go 2 Alex 1 Go 3 3 C++ 1 Python 2 Go 1

Выходные данные

YES
C++ 2 George
C++ 3 Andrew
Python 4 Peter
NO
YES
C++ 1 John
Python 2 Kate
Go 1 Mike

Н. Эвакуация в Берляндии

2 секунды, 256 мегабайт

Берляндия представляет собой n городов, соединёнными m односторонними дорогами. Берляндия всегда заботится о своих гражданах, в том числе на случай катастроф.

В k городах страны расположены современные убежища, где граждане могут переждать любое ненастье. Но так же есть довольно отдалённые города, из которых можно просто не успеть добраться до убежища.

Ваша задача — определить все города, жители которых успеют добраться до укрытия за время, не более d . Граждане городов, в которые расположены убежища, точно успеют добраться до укрытий.

Обратите внимание, что убежища очень вместительные, разрешается как всем ехать в одно убежище, так и всем в разные.

Входные данные

Первая строка входных данных содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов тестовых данных.

Первая строка набора тестовых данных содержит четыре целых числа n, m, k, d ($1 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5, 1 \leq k \leq \min(n, 10^4), 1 \leq d \leq 10^8$) — количество городов, дорог и убежищ в Берляндии, максимальное время, за которое требуется добраться до укрытия.

Вторая строка набора тестовых данных содержит k целых чисел a_i ($1 \leq a_i \leq n$) — количество городов, в которых размещено убежище.

Следующие m строк набора тестовых данных содержат по три целых числа u, v, w ($1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 10^6$) — номера городов, соединённые дорогой, и время, требуемое на проезд из u в v .

Гарантируется, что сумма n по всем наборам тестовых данных не превосходит $2 \cdot 10^5$ ($\sum n \leq 2 \cdot 10^5$).

Гарантируется, что сумма m по всем наборам тестовых данных не превосходит $5 \cdot 10^5$ ($\sum m \leq 5 \cdot 10^5$).

Выходные данные

На каждый набор тестовых данных выведите в единственной строке номера городов, из которых граждане успеют добраться до убежищ за время, не более d . Номера следует выводить в порядке возрастания.

Входные данные

3
11 14 2 5
1 9
2 1 8
4 1 4
5 4 2
5 3 1
3 1 6
5 6 2
3 6 1
6 9 7
8 9 2
10 9 10
11 10 1
7 11 2
11 8 3
2 8 4
11 14 2 6
1 9
2 1 8
4 1 4
5 4 2
5 3 1
3 1 6
5 6 2
3 6 1
6 9 7
8 9 2
10 9 10
11 10 1
7 11 2
11 8 3
2 8 4
11 14 2 7
1 9
2 1 8
4 1 4
5 4 2
5 3 1
3 1 6
8 9 2
10 9 10
11 10 1
7 11 2
11 8 3
2 8 4

Выходные данные

1 4 8 9 11
1 2 3 4 5 8 9 11
1 2 3 4 5 6 7 8 9 11

I. Самая ценная строка

3 секунды, 256 мегабайт

Максим начал изучать строковые суффиксные структуры и решил дать друзьям попробовать решить одну из задач. А ещё Максим считает, что короткие условия к задаче делают её привлекательной, поэтому в этой задаче оно короткое.

Вам дана строка s . Определим *ценность* подстроки t как произведение $|t| \cdot k$, где k — количество вхождений t в s .

Ваша задача — найти такую подстроку, у которой максимальная ценность.

Входные данные

Единственная строка входных данных содержит строку s длины n ($1 \leq n \leq 5 \cdot 10^3$). Строка состоит из строчных латинских букв.

Выходные данные

В единственной строке выведите строку t — ответ на задачу.

Входные данные

ababaaba

выходные данные
aba

входные данные
z
выходные данные
z

входные данные
ababab
выходные данные
abab

J. Да, это обязательно

2 секунды, 256 мегабайт

После неудачной стрелки с небезызвестным преподавателем Вы были отчислены и призваны в армию.

Товарищу полковнику понравилось, как Вы красите траву на поле, поэтому он поручил Вам новую боевую задачу — чистку снега! Плац представляет собой прямоугольник n на m метров. Изначально весь плац в снегу. Убрать весь снег сразу довольно трудно, поэтому Вам дают приказы последовательно — чистить по одному квадратному метру снега.

Представим плац как клетчатое поле n на m . Тогда приказ полковника — очистить клетку с координатами x и y . Теперь товарищ полковник решил Вам напомнить, для чего нужен плац — конечно же, для строевой подготовки!

Заниматься строевой подготовкой можно только на очищенных местах. Причём если очищены две соседние клетки, то торжественный марш проходит сразу по обоим клеткам. После каждой очистки снега товарищ полковник интересуется, в скольких местах возможно заниматься строевой подготовкой.

Входные данные

Товарищ полковник показывает Вам t плацев, которые Вам предстоит чистить ($1 \leq t \leq 10^4$).

Сначала товарищ полковник доводит до Вас три целых числа n, m, q ($1 \leq n, m \leq 2 \cdot 10^3, 1 \leq q \leq 2 \cdot 10^5$) — размеры очередного плаца и количество приказов.

Следующие q раз товарищ полковник отдаёт приказ в виде двух целых чисел x, y ($1 \leq x \leq n, 1 \leq y \leq m$) — координаты клетки плаца, которую нужно очистить.

Товарищ полковник не заставляет Вас чистить уже чистую клетку.

Гарантируется, что за весь срок службы Вам придётся чистить снег не более $2 \cdot 10^5$ раз ($\sum q \leq 2 \cdot 10^5$).

Выходные данные

Каждый раз, когда Вы очистили очередной квадратный метр плаца, доложите полковнику количество мест, где можно заниматься строевой подготовкой.

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