

Атака графов!

А. Катание на Коньках

2 seconds, 256 megabytes

Байтек учится кататься на льду. Он новичок, и поэтому он передвигается следующим образом: сначала отталкивается от сугроба на север, восток, юг или запад — и катится до тех пор, пока не повстречает очередной сугроб. Байтек заметил, что таким образом он не сможет добраться от некоторых сугробов до некоторых других, какой бы ни была последовательность его движений. Теперь он хочет соорудить несколько дополнительных сугробов так, чтобы он мог добраться от любого сугроба до любого другого. Байтек попросил Вас найти наименьшее количество сугробов, которые ему потребуется соорудить.

Мы предполагаем, что Байтек может сооружать сугробы только в точках с целочисленными координатами.

Входные данные

В первой строке входного файла записано единственное целое число n ($1 \leq n \leq 100$) — количество сугробов. Каждая из следующих n строк содержит по два целых числа x_i и y_i ($1 \leq x_i, y_i \leq 1000$) — координаты i -ого сугроба.

Обратите внимание, что направление на север совпадает с направлением оси Oy , таким образом, направление на восток совпадает с направлением оси Ox . Все сугробы расположены в различных точках.

Выходные данные

Выведите наименьшее количество сугробов, которые надо соорудить Байтеку для того, чтобы он мог добраться от любого сугроба до любого другого.

входные данные

2
2 1
1 2

выходные данные

1

входные данные

2
2 1
4 1

выходные данные

0

В. КПП

2 секунды, 256 мегабайт

В вашем городе n перекрестков. Между этими перекрестками расположены m **односторонних** дорог. Вы — мэр города, и вы должны обеспечивать безопасность на всех перекрестках.

Для обеспечения безопасности перекрестков нужно построить пункты КПП. КПП можно построить только на перекрестке. КПП на перекрестке i может обезопасить перекресток j , либо если $i = j$, либо если полицейская машина может доехать до j от i и затем вернуться в i .

Постройка КПП стоит денег. Так как некоторые районы города дороже других, то на некоторых перекрестках строить КПП дороже, чем на других. Какую минимальную сумму денег необходимо потратить для обеспечения безопасности всех перекрестков? Сколько существует оптимальных способов постройки КПП (**в дополнение к минимальной стоимости подстройки, вы должны построить как можно меньше КПП**)? Два способа считаются различными, если в одном из них некоторый перекресток имеет КПП, а в другом — нет.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 10^5$) — количество перекрестков. В следующей строке записано через пробел n целых чисел: i -е число равняется цене постройки КПП на i -м перекрестке (цена неотрицательная и не превышает 10^9).

В следующей строке записано целое число m ($0 \leq m \leq 3 \cdot 10^5$) — количество дорог. В каждой из следующих m строк записано по два целых числа, u_i и v_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$). Пара u_i, v_i означает, что в городе есть дорога, ведущая из перекрестка u_i в перекресток v_i . Гарантируется, что во входных данных между двумя перекрестками не может быть более одной дороги в одном направлении.

Считайте, что все перекрестки пронумерованы целыми числами от 1 до n .

Выходные данные

Выведите два целых числа через пробел. Первое число — минимально возможная сумма денег, необходимая для того, чтобы обеспечить безопасность всех перекрестков. Второе число — количество способов обеспечения безопасности. Так как количество способов может быть достаточно большим, выведите его по модулю 1000000007 ($10^9 + 7$).

входные данные

```
3
1 2 3
3
1 2
2 3
3 2
```

выходные данные

```
3 1
```

входные данные

```
5
2 8 0 6 0
6
1 4
1 3
2 4
3 4
4 5
5 1
```

выходные данные

```
8 2
```

входные данные

```
10
1 3 2 2 1 3 1 4 10 10
12
1 2
2 3
3 1
3 4
4 5
5 6
5 7
6 4
7 3
8 9
9 10
10 9
```

выходные данные
15 6

входные данные
2 7 91 2 1 2 2 1
выходные данные
7 1

С. Лиса и имена

2 секунды, 256 мегабайт

Лиса Ciel собирается опубликовать статью по ЛИС (Лисооперируемым Интеллектуальным Системам). Она слышала, что список авторов научной статьи всегда сортируется в лексикографическом порядке.

После изучения примеров оформления, лиса обнаружила, что иногда это утверждение не верно. В некоторых статьях имена авторов не сортируются в лексикографическом порядке в обычном понимании. Но, оказывается, верно то, что после некоторого изменения порядка букв в алфавите, порядок авторов становится лексикографическим!

Лиса хочет знать, существует ли такой порядок букв латинского алфавита, что имена авторов предлагаемой ею статьи следуют в лексикографическом порядке. Если да, то требуется также найти любой такой порядок.

Лексикографический порядок определяется следующим образом. Мы сравниваем s и t , сперва находя крайнюю левую позицию с различающимися символами: $s_i \neq t_i$. Если такой позиции нет (то есть, s — это префикс t или наоборот), то более короткая строка меньше. В противном случае мы сравниваем символы s_i и t_i согласно их порядку в алфавите.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 100$), количество имен.

В каждой из следующих n строк записано по одному слову $name_i$ ($1 \leq |name_i| \leq 100$), обозначающему i -е имя. Каждое имя содержит только строчные буквы латинского алфавита. Все имена различны.

Выходные данные

Если существует такой порядок букв, при котором имена в данном списке следуют в лексикографическом порядке, выведите любой такой порядок в виде перестановки символов 'a'-'z' (иными словами, выведите сначала первую букву модифицированного алфавита, затем вторую, и так далее).

В противном случае выведите единственное слово «Impossible» (без кавычек).

входные данные
3 rivest shamir adleman
выходные данные
bcdefghijklmnopqrsatuvwxyz

Входные данные
10 tourist petr wjmzmr yepitons vepifanov scottwu oooooooooooooooo subscriber rowdark tankengineer
Выходные данные
Impossible

Входные данные
10 petr egor endagorion feferivan ilovetanyaromanova kostka dmitriy maratsnowbear bredorjaguarturnik cgyforever
Выходные данные
aghjlnopefikmbcqrstuvwxyz

Входные данные
7 car care careful carefully becarefuldontforgetsomething otherwiseyouwillbehacked goodluck

Выходные данные
abcdefghijklmnopqrstuvwxyz

D. Цикл в графе

2 секунды, 256 мегабайт

Вам задан неориентированный граф G , состоящий из n вершин. Будем считать, что вершины этого графа пронумерованы целыми числами от 1 до n . Известно, что каждая вершина графа G соединена ребрами не менее чем с k другими вершинами этого графа. Ваша задача — найти в описанном графе простой цикл длины не менее $k + 1$.

Простым циклом длины d ($d > 1$) в графе G называется последовательность различных вершин графа v_1, v_2, \dots, v_d такая, что вершины v_1 и v_d соединены ребром графа, а также для любого целого i ($1 \leq i < d$) вершины v_i и v_{i+1} соединены ребром графа.

Входные данные

В первой строке записаны три целых числа n, m, k ($3 \leq n, m \leq 10^5$; $2 \leq k \leq n - 1$) — количество вершин в графе, ребер в графе и ограничение снизу на степень вершины графа. В следующих m строках записаны пары целых чисел. В i -той строке записаны числа a_i, b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$) — номера вершин графа, которые соединяет i -тое ребро.

Гарантируется, что заданный граф не содержит кратных ребер и петель. Гарантируется, что каждая вершина графа соединена ребрами не менее чем с k другими вершинами этого графа.

Выходные данные

В первой строке выведите целое число r ($r \geq k + 1$) — длина найденного цикла. В следующей строке выведите r различных целых чисел v_1, v_2, \dots, v_r ($1 \leq v_i \leq n$) — найденный простой цикл.

Гарантируется, что ответ существует. Если существует несколько правильных ответов, разрешается вывести любой из них.

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

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

Е. Егор и граф

3 секунды, 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из n вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из n шагов.
- На i -том шаге Егор удаляет из графа вершину номер x_i . Удаляя вершину, Егор удаляет все ребра, которые входили в данную

вершину и которые выходили из нее.

- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как $d(i, v, u)$ кратчайший путь между вершинами v и u в графе, который получился до удаления вершины x_i , то Егор хочет знать значение следующей суммы: $\sum_{v, u, v \neq u} d(i, v, u)$.

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Входные данные

В первой строке содержится целое число n ($1 \leq n \leq 500$) — количество вершин в графе.

В следующих n строках содержится по n целых чисел — матрица смежности графа: j -тое число в i -той строке a_{ij} ($1 \leq a_{ij} \leq 10^5$, $a_{ii} = 0$) обозначает вес ребра, ведущего из вершины i в вершину j .

В следующей строке содержится n различных целых чисел: x_1, x_2, \dots, x_n ($1 \leq x_i \leq n$) — вершины, которые удаляет Егор.

Выходные данные

Выведите n целых чисел — i -тое число равно искомой сумме перед i -тым шагом.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на C++. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

входные данные
1 0 1
выходные данные
0

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

входные данные
4 0 3 1 1 6 0 400 1 2 4 0 1 1 1 1 0 4 1 2 3
выходные данные
17 23 404 0

Ф. Дима и бактерии

2 секунды, 256 мегабайт

Дима стал заниматься биологией бактерий, в результате своих опытов он вывел k видов бактерий. Всего сейчас в его лаборатории содержится n бактерий, причем количество бактерий вида i равно c_i . Для удобства будем считать, что все бактерии пронумерованы от 1 до n . При этом бактерии вида c_i имеют номера от $(\sum_{k=1}^{i-1} c_k) + 1$ до $\sum_{k=1}^i c_k$.

С помощью специального оборудования Дима может передавать энергию между некоторыми бактериями. Конечно, пользование таким оборудованием не бесплатно. Дима знает m способов передачи энергии с помощью оборудования. Способ с номером i характеризуется тремя целыми числами v_i, u_i, x_i и позволяет за x_i долларов передавать энергию от бактерии с номером u_i к бактерии с номером v_i или наоборот.

Начальник Димы (Инна) считает, что распределение на виды устроено *правильным образом*, если существует способ (возможно, использовав оборудование несколько раз) передать энергию от любой бактерии определенного вида другой бактерии этого же вида бесплатно (для любых двух бактерий одного и того же вида).

Поскольку в таком случае стоимость передачи энергии от любой бактерии к любой другой зависит лишь только от видов этих двух бактерий, помогите Инне определить: образуют ли выведенные Димой бактерии устроенное правильным образом распределение на виды? И если образуют, выведите матрицу d размера $k \times k$. Ячейка $d[i][j]$ этой матрицы должна обозначать стоимость передачи энергии от бактерии вида i к бактерии вида j .

Входные данные

В первой строке записаны три целых числа n, m, k ($1 \leq n \leq 10^5$; $0 \leq m \leq 10^5$; $1 \leq k \leq 500$). В следующей строке записаны k целых чисел c_1, c_2, \dots, c_k ($1 \leq c_i \leq n$). В каждой из следующих m строк записано три целых числа u_i, v_i, x_i

($1 \leq u_i, v_i \leq n$; $0 \leq x_i \leq 10^4$; $u_i \neq v_i$). Гарантируется, что $\sum_{i=1}^k c_i = n$.

Выходные данные

Если распределение Димы устроено правильным образом, выведите строку «Yes», а затем k строк: в i -й строке выведите целые числа $d[i][1], d[i][2], \dots, d[i][k]$ ($d[i][i] = 0$). Если нельзя передать энергию бактерии типа j от бактерии типа i соответствующее значение $d[i][j]$ должно быть равно -1. Если распределение Димы устроено неправильным образом, выведите «No».

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

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

входные данные
3 2 2 2 1 1 2 0 2 3 1
выходные данные
Yes 0 1 1 0

входные данные
3 0 2 1 2

выходные данные
No

G. Воссоединение семьи

3 секунды, 512 мегабайт



Сиджей привез своего брата Свита на встречу с представителями авторитетных семей Гроув Стрит. В силу договоренностей между участниками встречи, авторитетный брат пошел туда один, а Сиджей остался в машине его дожидаться.

Спустя несколько минут, Сиджей заметил, как появился спецназ и начал штурмовать отель. Сиджей решил ворваться и спасти своего брата, но перед этим, конечно же, построить оптимальный маршрут.

Отель, где проходит встреча, представляет из себя прямоугольное клетчатое поле, состоящее из N строк и M столбцов. Клетки, обозначенные как #, являются непроходимыми. Все остальные клетки - проходимые. Текущая позиция Сиджея на карте обозначена как C , а текущая позиция Свита - S .

Из текущей клетки Сиджей может перемещаться в четырех направлениях: влево, вправо, вверх и вниз, переходя в одну из соседних по стороне клеток, если она является проходимой. На каждое такое перемещение он тратит 1 секунду.

К счастью, Сиджей знает расположение на карте реактивных ранцев **R**. Добравшись до реактивного ранца и надев его, Сиджей может передвигаться в текущем направлении в **любую клетку на пути по прямой** до ближайшей непроходимой клетки без потери времени, но поворот в текущей клетке теперь занимает 1 секунду. Сиджей не может снять однажды надетый реактивный ранец. Если он переходит в клетку с ранцем, то ранец надевается автоматически, при этом текущее направление движения сохраняется.

Заметьте, что при повороте в реактивном ранце позиция Сиджея не изменяется, меняется только текущее направление движения. Если Сиджей надел реактивный ранец и движется в текущем направлении, то поворот направо или налево занимает у него 1 секунду, а разворот в противоположном направлении уже 2, так как это два поворота направо (или два поворота налево). Поворот без реактивного ранца выполняется без потери времени.

Помогите ему вычислить величину кратчайшего пути до Свита. Запрещено: перелетать непроходимые клетки; выходить за пределы карты; использовать читы.

Входные данные

В первой строке находятся два целых числа N и M - количество строк и столбцов, соответственно ($2 \leq N, M \leq 1000$).

В каждой из следующих N строк находится M символов - описание i -й строки карты. Гарантируется, что среди всех строк буквы **C** и **S** встречаются ровно один раз каждая.

Реактивных ранцев может быть сколько угодно.

Выходные данные

Выведите единственное число - величину кратчайшего пути до Свита. Если до Свита добраться невозможно, выведите -1 .

входные данные

```
7 7
.....C
.#####
.....R
.#####
.#S....
.#####.
.....
```

выходные данные

```
20
```

входные данные

```
7 7
.....C
.#####
.....R
.#####
.#S....
.##....
.....
```

выходные данные

```
18
```

В первом тестовом примере выгодно взять рюкзак, потратив на это 14 секунд. После взятия окажется, что нужно развернуться в противоположном направлении - это два поворота направо на месте (еще 2 секунды). Затем нужно долететь по прямой до клетки (3, 1), в этой клетке повернуть налево (+1 секунда), долететь до клетки (7, 1), повернуть налево (+1 секунда), и еще два поворота налево в клетках (7, 7) и (5, 7). Итого:
 $14 + 2 + 1 + 1 + 1 + 1 = 14 + 6 = 20$ секунд.

Во втором тестовом примере Сиджею выгодно не брать рюкзак, а бежать сразу к Свиту напрямую. От **C** до **S** ровно 18 секунд, тогда как если взять рюкзак, то получатся все те же 20 секунд.

Н. Разделение графа

1 секунда, 256 мегабайт

Дан взвешенный неориентированный граф G , содержащий n вершин и m ребер.

Разбейте вершины графа на два **не пустых** множества так, чтобы ребро минимального веса, соединяющее вершины из одного множества, имело максимально большой вес.

Гарантируется, что граф не содержит петель и его нельзя разбить так, чтобы такого ребра не существовало.

Входные данные

В первой строке задано число вершин n ($3 \leq n \leq 10^5$) и число ребер m ($3 \leq m \leq 10^5$) графа.

В следующих m строках заданы ребра в формате u, v, w ($1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 10^5$), где u и v задают начальную и конечную вершину ребра, а w — его вес.

Выходные данные

В единственной строке выведите максимальный возможный вес ребра, которое имеет минимальный вес среди тех, что соединяют вершины, принадлежащие одному множеству.

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

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

Рассмотрим первый пример из условия. В нём возможно всего 3 различных разбиения, удовлетворяющих условию, рассмотрим каждое из них:

- $\{\{1, 2\}, \{3\}\}$, в таком случае рёбра 1 и 2 будут соединять вершины из одного множества, то есть ответ 1;
- $\{\{1, 3\}, \{2\}\}$, в таком случае только ребро 3 будет соединять вершины из одного множества, то есть ответ 3;
- $\{\{2, 3\}, \{1\}\}$, в таком случае только ребро 4 будет соединять вершины из одного множества, то есть ответ 4.

Получаем, что ответ на тест достигается при третьем варианте разбиения.

Во втором тесте ответ достигается на разбиении $\{\{1, 3\}, \{2, 4\}\}$. Легко убедиться, расписав все возможные разбиения, что не существует разбиения данного графа, на котором ответ был бы больше.

[Codeforces](#) (с) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0