

## Задача А. Колода для фокусов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Недавно были выпущены колоды игральных карт нового формата. Значение каждой карты в колоде задается `base64`-строкой длины  $k$ . В рамках данной задачи под `base64`-строкой подразумевается строка, каждый символ которой может принимать одно из следующих 64 значений: ‘#’, ‘\$’, ‘0’–‘9’, ‘A’–‘Z’ и ‘a’–‘z’.

Каждая колода состоит из  $n = 64^k$  карт и содержит ровно по одной карте с каждым возможным значением. Более того, в запечатанной новой колоде карты расположены в порядке лексикографического возрастания их значений. Напомним, что строка  $s$  лексикографически *больше* строки  $t$  тогда и только тогда, когда для некоторого  $i$  верно, что  $s_1 = t_1, \dots, s_{i-1} = t_{i-1}$ , а  $s_i > t_i$ . Иными словами, когда после их общего (возможно, пустого) префикса в  $s$  идет больший символ, чем в  $t$ .

Фокусник Чоно Урёку перемешивает колоду, бесконечное количество раз повторяя следующее действие:

1. сначала он разделяет колоду на две половины — в первой оказываются карты, находившиеся на позициях от 1 до  $\frac{n}{2}$ , во второй — от  $\frac{n}{2} + 1$  до  $n$  (порядок карт не меняется);
2. затем он собирает колоду заново, строго чередуя карты из двух половин, начиная с первой: если до перемешивания пронумеровать карты от 1 до  $n$ , то после перемешивания колода будет выглядеть следующим образом:

$$\left\langle 1, \frac{n}{2} + 1, 2, \frac{n}{2} + 2, \dots, \frac{n}{2}, n \right\rangle.$$

Чоно собирается показать  $m$  фокусов. Чтобы  $i$ -й фокус удался, ему необходимо, чтобы карта со значением  $x_i$  оказалась на позиции, на которой в изначальной (запечатанной) колоде находилась карта со значением  $y_i$ . Считая, что для каждого фокуса Чоно берет новую запечатанную колоду, помогите ему определить, какое минимальное число перемешиваний понадобится для каждого фокуса.

### Формат входных данных

В первой строке ввода через пробел даны два целых числа  $k$  и  $m$  — длина значения каждой карты и количество фокусов ( $1 \leq k \leq 2500$ ;  $1 \leq m \leq 1000$ ).

В следующих  $m$  строках перечислены описания фокусов. В  $i$ -й из них через пробел даны две строки  $x_i$  и  $y_i$  — значения карт, фигурирующих в  $i$ -м фокусе ( $|x_i| = |y_i| = k$ ). Гарантируется, что  $x_i$  и  $y_i$  состоят только из указанных в условии символов.

### Формат выходных данных

Для каждого фокуса выведите в отдельной строке номер перемешивания, после которого карта со значением  $x_i$  впервые окажется на позиции  $y_i$ .

Если карта изначально находится на нужной позиции, выведите 0. Если же такое событие никогда не произойдет, выведите  $-1$ .

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	8	$k = 1, m \leq 200$	—	полная
2	11	$k \leq 10$ , все $x_i$ равны между собой	—	полная
3	13	$k \leq 2$	1	полная
4	12	$k \leq 50, m \leq 200$	—	полная
5	18	$k, m \leq 300$	4	полная
6	19	$k = 1024, m \leq 700$	—	первая ошибка
7	19	нет	1 – 6	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
1 7 # # U \$ 4 3 t D D t 2 \$ \$ 2	0 1 -1 3 3 4 2
3 8 Hd7 CYZ mZ\$ 1Z8 iYq poa J1P edh SyR uxw aCp n50 I#g 0q8 wrP tir	2 13 15 -1 13 17 -1 1

## Замечание

ASCII коды используемых в значениях карт символов принимают значения 35 ('#'), 36 ('\$'), от 48 до 57 (цифры), от 65 до 90 (заглавные латинские буквы) и от 97 до 122 (строчные латинские буквы).

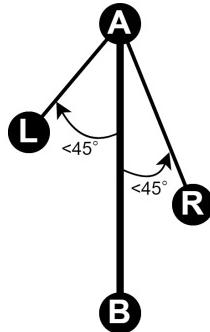
## Задача В. Гигантский дракон

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Канеки смотрит на неориентированный граф на плоскости из  $n$  вершин и  $m$  ребер. В этом графе ему интересно найти самого большого дракона.

Назовем *сегментом* дракона три ребра графа  $AL$ ,  $AB$  и  $AR$ , имеющие общую вершину  $A$ , и обладающие следующими свойствами:

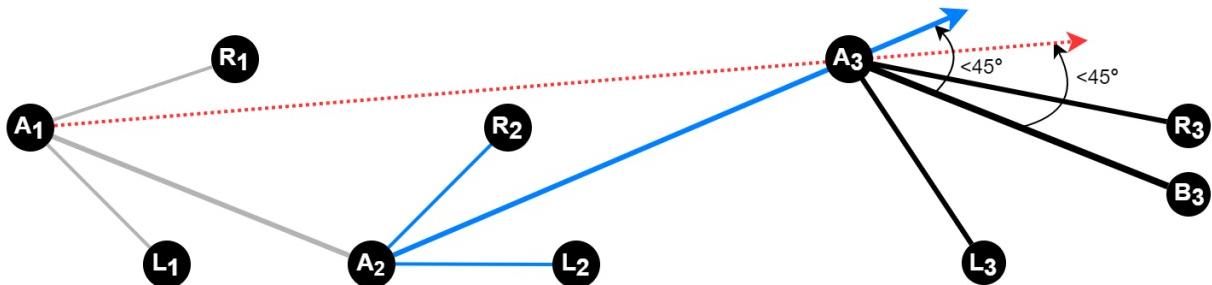
- $0 < \angle(BAL) < 45^\circ$  и направление поворота от  $\overrightarrow{AB}$  к  $\overrightarrow{AL}$  — **по** часовой стрелке;
- $0 < \angle(BAR) < 45^\circ$  и направление поворота от  $\overrightarrow{AB}$  к  $\overrightarrow{AR}$  — **против** часовой стрелки;
- $|AB| \geq |AL|$  и  $|AB| \geq |AR|$ , то есть  $AB$  — максимальное по длине из трех ребер.



При выполнении всех указанных условий вершины  $A$  и  $B$  называются *началом* и *концом* сегмента, а ребра  $AL$ ,  $AB$  и  $AR$  — *левой лапой*, *основанием* и *правой лапой* сегмента, соответственно.

Определим *дракона* как последовательность сегментов, в которой

- начало первого сегмента  $A_1$ , также называемое *головой* дракона, находится в вершине  $S$ ;
- $A_i = B_{i-1}$  для всех  $i > 1$ , то есть начало каждого следующего сегмента совпадает с концом предыдущего;
- $|\angle(\overrightarrow{A_{i-1}B_{i-1}}, \overrightarrow{A_iB_i})| < 45^\circ$ , то есть угол между векторами оснований соседних сегментов строго меньше  $45^\circ$ ;
- $|\angle(\overrightarrow{A_1A_i}, \overrightarrow{A_iB_i})| < 45^\circ$ , то есть угол между вектором от головы дракона  $A_1$  до начала сегмента и основанием сегмента строго меньше  $45^\circ$ .



Обратите внимание, что здесь углы взяты по модулю, то есть каждый следующий сегмент может быть повернут относительно предыдущего на менее чем  $45^\circ$  как по, так и против часовой стрелки.

Мощностью дракона будем считать сумму **квадратов** длин оснований его сегментов, то есть  $\sum |A_iB_i|^2$ . В заданном графе помогите Канеки найти дракона максимальной мощности с головой в вершине  $S$ .

## Формат входных данных

В первой строке входных данных даны три числа  $n, m, S$  ( $2 \leq n \leq 2 \cdot 10^5$ ;  $1 \leq m \leq 4 \cdot 10^5$ ;  $1 \leq S \leq n$ ) — количество вершин и ребер в заданном графе и номер вершины, являющейся головой дракона.

В следующих  $n$  строках дано описание вершин графа. Каждая строка содержит два целых числа  $x_i$  и  $y_i$  — координаты  $i$ -й вершины ( $0 \leq x_i, y_i \leq 10^9$ ). Гарантируется, что все вершины графа различны, то есть не существует двух вершин, обе координаты которых совпадают.

Далее следует пустая строка.

В следующих  $m$  строках дано описание ребер графа. Каждая строка содержит два целых числа  $u_i$  и  $v_i$  — номера вершин, соединенных  $i$ -м ребром ( $1 \leq u_i, v_i \leq n$ ;  $u_i \neq v_i$ ). Гарантируется, что граф не содержит кратных ребер.

## Формат выходных данных

В первой строке выходных данных выведите два числа  $k$  и  $ans$  — количество сегментов в драконе, имеющем максимальную мощность, и само значение его мощности.

В следующих  $k$  строках выведите описание сегментов в том порядке, в котором они образуют дракона. В качестве описания сегмента  $i$  выведите номера вершин  $L_i$ ,  $B_i$  и  $R_i$ .

Будем считать, что дракон может состоять только из вершины  $S$ . В таком случае количество сегментов и его мощность следует считать нулями.

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

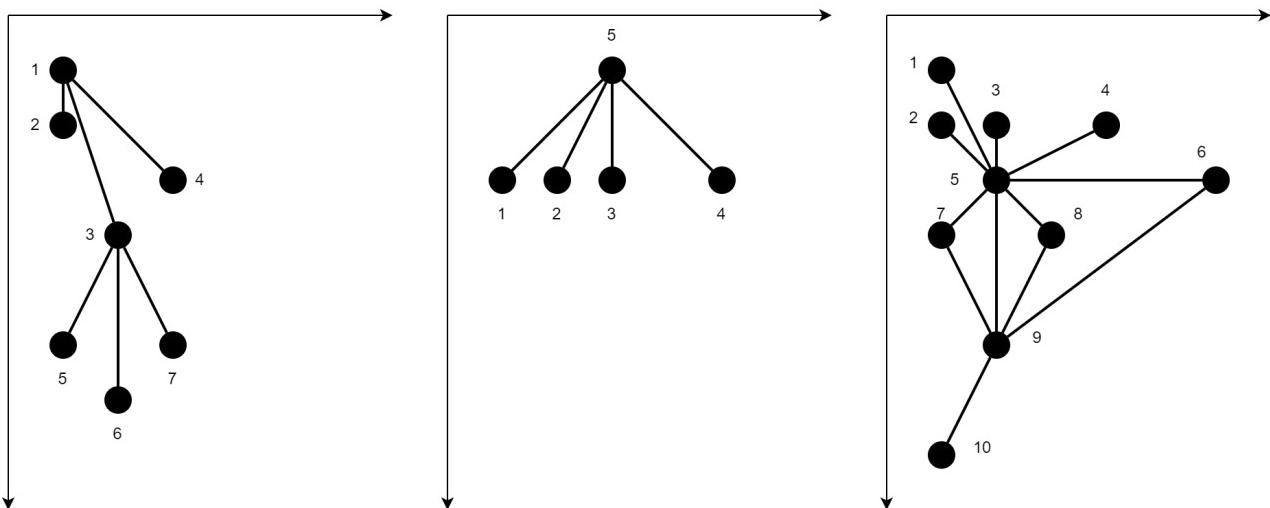
Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	12	$n, m \leq 5$		полная
2	18	$n, m \leq 10$	1	полная
3	16	граф — дерево		полная
4	16	$0 \leq x_i \leq 1$ для всех $i$		полная
5	18	$n \leq 10^3$ , $m \leq 10^4$	2	первая ошибка
6	20	нет	1 – 5	первая ошибка

**Примеры**

стандартный ввод	стандартный вывод
7 6 1 1 1 1 2 2 4 3 3 1 6 2 7 3 6  1 2 1 3 1 4 3 5 3 6 3 7	2 19 2 3 4 5 6 7
5 4 5 1 3 2 3 3 3 5 3 3 1  1 5 2 5 5 3 4 5	0 0
10 12 9 1 1 1 2 2 2 4 2 2 3 6 3 1 4 3 4 2 6 1 8  1 5 2 5 3 5 4 5 5 6 5 9 6 9 7 9 8 9 9 10 5 8 5 7	2 14 8 5 7 3 1 2

## Замечание

Графы, данные в первом, втором и третьем тесте условий, выглядят следующим образом.



- В первом тесте в качестве максимального дракона можно взять весь граф целиком;
- Во втором тесте ни одна тройка ребер не может быть взята в сегмент, так как не выполняется одно из обязательных условий;
- В третьем тесте максимальный дракон состоит из двух сегментов с основаниями  $9 \rightarrow 5$  и  $5 \rightarrow 1$  с лапами ( $9 \rightarrow 8, 9 \rightarrow 7$ ) и ( $5 \rightarrow 3, 5 \rightarrow 2$ ).

## Задача С. За Орду!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Страна состоит из  $N$  городов, пронумерованных от 1 до  $N$ . Эти города могут заключать договоры, которые означают их готовность сотрудничать друг с другом.

Альянс — это группа городов, где каждый город соединен с каждым другим городом в группе через один или несколько договоров. Размер альянса — это количество городов, входящих в этот альянс.

Каждый договор имеет длину, представляющую количество страниц, использованных для его составления. Чтобы сократить бюрократию, города иногда удаляют самый длинный активный договор (то есть тот, у которого наибольшее количество страниц).

Помимо административных вопросов, города также проводят раунды войн между альянсами. Война следует следующим правилам:

- Каждый альянс формирует команду.
- Альянсы объединяются в пары для игры друг против друга.
- Если количество альянсов нечетное, одна команда будет играть против пустого альянса размера 0.
- Показатель несправедливости матча между двумя альянсами размеров  $A$  и  $B$  задается как:  $|A - B|$
- Общий показатель несправедливости для раунда — это сумма показателей несправедливости по всем матчам.

Вы должны определить минимально возможный общий показатель несправедливости, оптимально объединяя альянсы в пары.

Вы должны обработать  $Q$  запросов следующих типов:

- «а»: Добавить договор между городами  $u$  и  $v$  с  $p$  страницами.
- «г»: Удалить самый длинный договор (то есть тот, у которого наибольшее количество страниц).

Все количества страниц уникальны.

- «д»: Вычислить минимальный показатель несправедливости для войны на основе текущих альянсов.

Запросы типа «д» должны быть обработаны немедленно по запросу, до обработки следующих запросов.

Ограничения

- $1 \leq N \leq 100000$ .
- $1 \leq Q \leq 500000$ .
- $1 \leq p \leq 10^9$ .
- $1 \leq u \leq N$ .
- $1 \leq v \leq N$ .
- $u \neq v$  для каждого запроса типа «а».
- При добавлении договора между  $u$  и  $v$ , непосредственно перед этим не действует никакой договор между  $u$  и  $v$ .
- Все количества страниц  $p$  уникальны.

### Формат входных данных

Ваша программа должна читать стандартный ввод в следующем формате:

строка 1:  $N \ Q$

Следующие  $Q$  строк имеют один из следующих форматов:

- «а»  $u \ v \ p$
- «г»
- «д»

Соответствующие различным операциям-запросам.

## Формат выходных данных

Ваша программа должна для каждого запроса типа «d» немедленно отвечать перед обработкой любых дополнительных запросов.

Кроме того, вы должны сбрасывать буфер вывода немедленно после каждого ответа.

## Система оценки

Тесты к этой задаче состоят из 7 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп.

1. (9 баллов)  $N \leq 10, Q \leq 20$ .
2. (10 баллов)  $N \leq 2000, Q \leq 4000$ .
3. (6 баллов) Не более 10 запросов «d».
4. (17 баллов) Для каждого запроса «a»,  $u + 1 = v$ .
5. (14 баллов) Договоры создаются в порядке возрастания количества страниц.
6. (26 баллов) Договоры создаются в порядке убывания количества страниц.
7. (18 баллов) Без дополнительных ограничений.

## Примеры

стандартный ввод	стандартный вывод
3 5 a 1 2 1 a 2 3 2 d r d	3 1
6 10 a 2 3 10 a 1 2 5 a 3 4 8 d r d a 4 5 1 a 3 6 7 r d	4 0 2

## Замечание

Когда проводятся первая война, города 1, 2 и 3 находятся в одном альянсе. Так что они играют против пустого альянса с показателем несправедливости  $|0-3| = 3$ . Затем договор между городом 2 и городом 3 удаляется. Так что турнир по вышибалам проходит между альянсом из городов 1 и 2 против альянса города 3. Это дает показатель несправедливости  $|2-1|=1$ .

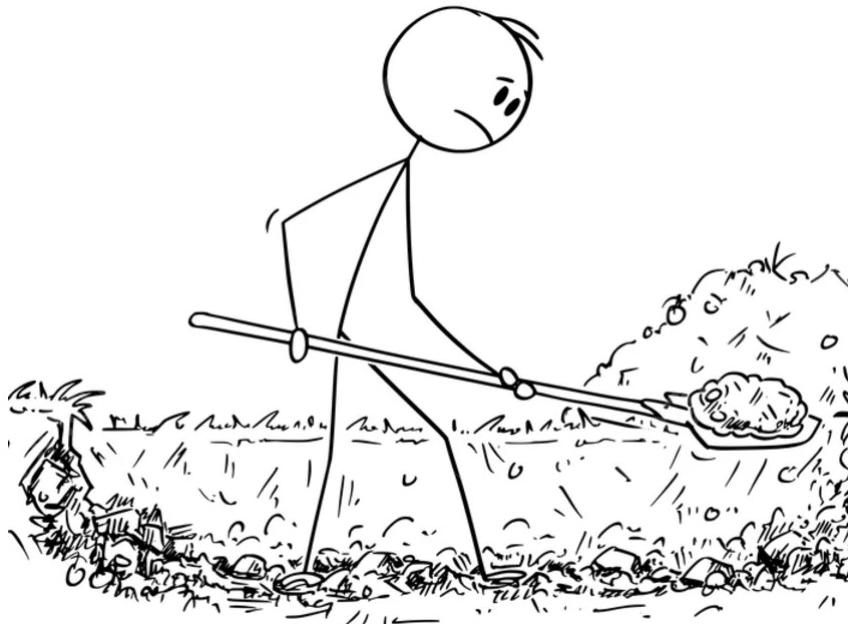
Для первого раунда войны есть один альянс размера 4 и два альянса размера 1. Это дает общий показатель несправедливости 4.

Для второго раунда вышибал есть два альянса размера 2 и два альянса размера 1, с общим показателем несправедливости 0.

Для третьего раунда вышибал есть три альянса размера 2, с общим показателем несправедливости 2.

## Задача D. Карьер

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 5 секунд  
 Ограничение по памяти: 512 мегабайт



Недалеко от города N обнаружили богатое месторождение бурого угля. Месторождение в разрезе представляет собой клетчатую полуплоскость, в  $n$  клетках которой находятся запасы угля: в клетке с координатой  $(x_i, y_i)$  (где  $x_i$  — координата по горизонтали,  $y_i$  — глубина) залегают запасы угля стоимостью  $w_i$  тугриков. Чтобы добыть как можно больше угля, было принято решение выкопать карьер. Чтобы карьер не обрушился, было решено сначала выкопать целиком некоторый отрезок клеток  $[l_1, r_1]$  на глубине 1, затем **строго вложенный** в него отрезок  $[l_2, r_2]$  ( $l_1 < l_2 \leq r_2 < r_1$ ) на глубине 2, отрезок  $[l_3, r_3]$  на глубине 3 и так далее до выбранной максимальной глубины  $h$ .

Таким образом, план карьера представляет собой два массива:  $l_1, l_2, \dots, l_h$  и  $r_1, r_2, \dots, r_h$ . При этом должно выполняться неравенство  $l_1 < l_2 < \dots < l_h \leq r_h < \dots < r_2 < r_1$ . Все клетки  $(x, y)$ , такие, что  $y \leq h$  и  $l_y \leq x \leq r_y$ , считаются выкопанными. Однако строительство карьера — затратное дело, поэтому за каждую выкопанную клетку нужно заплатить 1 тугрик вне зависимости от наличия в ней угля. Итоговая прибыль от карьера равна суммарной стоимости угля во всех выкопанных клетках за вычетом общего количества выкопанных клеток.

Составьте план карьера таким образом, чтобы прибыль от него была максимально возможной, или сообщите, что карьера с положительной прибылью не существует.

### Формат входных данных

В первой строке вводится одно целое число  $n$  — количество залежей угля ( $1 \leq n \leq 10^5$ ).

В следующих  $n$  строках вводится по 3 целых числа —  $x_i$ ,  $y_i$  и  $w_i$  — координаты  $i$ -й залежи и стоимость угля в ней ( $1 \leq x_i \leq 10^9$ ,  $1 \leq y_i \leq 10^6$ ,  $1 \leq w_i \leq 10^9$ ).

### Формат выходных данных

В первой строке выведите одно целое число — максимальную возможную прибыль от карьера.

В следующих строках выведите план карьера, прибыль от которого максимальна.

Во второй строке выведите одно целое число  $h$  — глубину карьера.

В следующих  $h$  строках выведите по 2 целых числа  $l_i$  и  $r_i$  ( $l_i \leq r_i$ ) — границы отрезка карьера на глубине  $i$ .

Если любой карьер с ненулевой глубиной имеет отрицательную прибыль, вместо ответа выведите 0.

## Система оценки

В этой задаче 10 групп тестов. В группах 1 – 5 баллы начисляются за каждый пройденный тест. Баллы за группы 6 – 10 начисляются только при прохождении всех тестов группы и необходимых групп.

Подзадача	Баллы	Доп. ограничения	Система оценки	Необх. группы
1	10	$x_i = 1$	Потестовая	–
2	10	$y_i = 1$	Потестовая	–
3	10	$x_i \leq 10^4, y_i \leq 10^4, w_i = 10^9$	Потестовая	–
4	10	$x_i \leq 100, y_i \leq 100$	Потестовая	–
5	10	$n \leq 200$	Потестовая	–
6	10	$x_i \leq 2000, y_i \leq 2000$	Полная группа	4
7	10	$n \leq 2000$	Полная группа	5
8	10	$y_i \leq 2000$	Полная группа	4, 6
9	10	$y_i \leq 30000$	Полная группа	4, 6, 7
10	10	–	Полная группа	1 – 9

## Пример

стандартный ввод	стандартный вывод
9 2 2 4 2 4 3 4 3 4 5 1 4 6 3 4 3 2 1 4 2 2 1 3 1 6 2 1	4 3 1 6 2 5 4 4

## Замечание

В первом примере максимальная прибыль, которую можно получить — 4 тугрика. Для этого нужно выкопать отрезок  $[1, 6]$  на глубине 1, отрезок  $[2, 5]$  на глубине 2, и отрезок  $[4, 4]$  на глубине 3. Суммарная стоимость выкопанного угля равна  $4 + 4 + 1 + 2 + 4 = 15$ , при этом всего выкопано  $6 + 4 + 1 = 11$  клеток, поэтому итоговая прибыль составляет  $15 - 11 = 4$  тугрика.

