

## Задача D. Простая игра

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

НурлашКО хорошо вел себя в прошедшем году, и за это Дед Мороз подарил ему на Новый год ломаную цепь из  $N$  вершин.  $i$ -ая вершина ломаной находится в точке с координатами  $(i, y_i)$ .

Почти сразу была выпущена новая игра с этой геометрической фигурой:  $M$  раз выполняется одна из следующих операций:

- Изменить координату  $y$  одной из вершин ломаной.
- Провести горизонтальную линию на высоте  $H$  и подсчитать сколько раз она пересечется с ломаной. Отметим, что все точки горизонтальной линии имеют  $y$  координату равной  $H$ .

НурлашКО очень понравилась эта игра и он просит Вас написать для нее программу.

### Формат входных данных

Первая строка входного файла содержит два целых положительных числа  $N, M$  ( $1 \leq N, M \leq 100\,000$ ) — количество вершин ломаной и количество операций в игре, соответственно.

Следующая строка содержит  $N$  целых положительных чисел разделенных единичным пробелом  $h_i$  ( $1 \leq h_i \leq 1\,000\,000$ ) —  $h_i$  есть изначальная высота  $i$ -ой вершины.

Далее в  $M$  строках идут описания операций в игре в следующем формате:

- $1\ pos\ val$  ( $1 \leq pos \leq N, 1 \leq val \leq 1\,000\,000$ ) — номер вершины и новая высота, соответственно.
- $2\ H$  ( $1 \leq H \leq 1\,000\,000$ ) — высота горизонтальной прямой. Гарантируется что эта прямая никогда не проходит через вершины ломаной.

### Формат выходных данных

Для каждого запроса второго типа, в отдельной строке, выведите количество пересечений ломаной с горизонтальной прямой. Ответы на запросы выводите в том же порядке в котором они идут во входном файле.

### Система оценки

Данная задача содержит 3 подзадачи:

1.  $1 \leq N, M \leq 1\,000$ . Оценивается в 22 баллов.
2.  $1 \leq N, M \leq 100\,000$ . Разрешены только запросы (операции второго типа). Оценивается в 27 баллов.
3.  $1 \leq N, M \leq 100\,000$ . Оценивается в 51 баллов.

Каждая следующая подзадача оценивается только при прохождении всех предыдущих.

### Пример

<code>game.in</code>	<code>game.out</code>
3 3	2
1 5 1	1
2 3	
1 1 5	
2 3	

## Задача Е. Бомба

Имя входного файла: `bomb.in`  
Имя выходного файла: `bomb.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 128 мегабайт

Земляне и инопланетяне ведут бой за Марс. Бой идет на прямоугольном клеточном поле размером  $N \times M$ . Каждая клетка полностью принадлежит одной из сторон. Земляне могут создать бомбу, поражающую каждую клетку некоей прямоугольной области на поле боя, стороны которых параллельны сторонам поле боя. Бомбу нельзя поворачивать и использовать за пределами поля боя. Бомбу можно использовать неограниченное количество раз. Естественно, земляне не хотят, чтобы бомба попала на свои клетки, но они могут создать бомбу только одного определенного размера. Вычислите максимальную площадь поражения (т.е. произведению высоты и ширины) бомбы, используя которую можно поразить все клетки инопланетян, а клетки Землян останутся целыми. Любую клетку инопланетян можно поразить многократно.

### Формат входных данных

В первой строке находятся два целых числа  $N, M$  ( $1 \leq N, M \leq 2500$ ), разделенных пробелом, где  $N$  и  $M$  — соответственно высота и ширина поля боя. Далее следуют  $N$  строк по  $M$  символов каждая, задающие поле боя. Если символ в строке равен «0», то соответствующая клетка принадлежит Землянам, а если он равен «1», то эта клетка принадлежит инопланетянам.

### Формат выходных данных

Выведите одно число - площадь области поражения бомбы.

### Система оценки

В данной задаче ровно 100 тестов:

1. В тестах 1-6:  $N = 1$  или  $M = 1$ .
2. В тестах 7-16:  $1 \leq N, M \leq 20$ .
3. В тестах 17-26:  $1 \leq N, M \leq 100$ .
4. В тестах 27-36:  $1 \leq N, M \leq 450$ .
5. В тестах 37-100:  $1 \leq N, M \leq 2500$ .

За каждый пройденный тест участник получает 1 балл.

### Пример

<code>bomb.in</code>	<code>bomb.out</code>
5 6 001000 111110 111110 111110 000100	3

### Замечание

В тестовом примере размер оптимального прямоугольника равен  $3 \times 1$ .

## Задача F. Крутой маршрут

Имя входного файла:	road.in
Имя выходного файла:	road.out
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Мансур — является правителем страны АСМстан. В этой стране  $N$  городов и  $N - 1$  двусторонних дорог. Известно что, с каждого города можно добраться до любого другого двигаясь по существующим дорогам. Более формально: страна выглядит как дерево, где вершинами являются города, а ребра двусторонние дороги.

Также, в этой стране города из которых выходит **ровно одна** дорога называются *конечными*. Маршрутом называется простой путь от одного *конечного* города до другого *конечного* города. Расстояние между двумя городами есть минимальное число дорог которое требуется пройти чтобы идти от одного города до другого. Расстоянием от какого-либо города до маршрута есть минимальное количество дорог, которое требуется пройти от заданного города до какого-либо города на маршруте. Мансур решил внедрить в город **ровно один** маршрут, но Мансур заинтересован только в *крутых* маршрутах. Крутость маршрута считается следующим способом: пусть  $A$  и  $B$  это *конечные* города данного маршрута, а  $H$  это максимальное расстояние от какого-либо города в стране до этого маршрута, тогда *крутость* данного маршрута это произведение расстояния между  $A$  и  $B$  на  $H$ .

Мансур дал задание Темирулану найти максимальную *крутость* среди всех маршрутов, более того ему интересно знать количество таких маршрутов. Темирулан просит помощи у Вас.

Настоятельно рекомендуем прочесть пояснение к примеру.

### Формат входных данных

В первой строке входных данных содержится целое положительное число  $N$  ( $2 \leq N \leq 500000$ ) — количество городов в стране. Города пронумерованы от 1 до  $N$ . В следующих  $N - 1$  строках содержится по 2 целых положительных числа, разделенных пробелом,  $u_i, v_i$  ( $1 \leq u_i, v_i \leq N; u_i \neq v_i$ ) — дорога соединяющая города  $u_i$  и  $v_i$ . Гарантируется, что заданный граф дерево.

### Формат выходных данных

В единственной строке выходных данных выведите два целых числа — максимальную *крутость* и количество маршрутов, разделенные пробелом. **Обратите внимание**, что маршрут от  $A$  до  $B$  и маршрут от  $B$  до  $A$  считаются **одним и тем же** маршрутом.

### Система оценки

Данная задача содержит три подзадачи:

1.  $2 \leq N \leq 100$ . Оценивается в 19 баллов.
2.  $2 \leq N \leq 5000$ . Оценивается в 33 баллов.
3.  $2 \leq N \leq 500000$ . Оценивается в 48 баллов.

Каждая подзадача оценивается только при прохождении всех предыдущих.

## Примеры

road.in	road.out
7 1 2 1 3 2 4 2 5 3 6 3 7	6 2
4 1 2 2 3 2 4	2 3
5 1 2 2 3 3 4 4 5	0 1

## Замечание

Путь называется простым, если вершины в нем не повторяются. Обратите внимание, что не все простые пути являются *маршрутом*.

В первом тестовом примере:

Четыре *конечных* города с номерами 4, 5, 6 и 7. Если выбрать маршрутом путь 4-2-1-3-6, то расстояние между ними равно 4 и расстояния от других городов до этого маршрута равны [1, 1], максимальное среди них 1, соответственно *крутость* маршрута равна  $4 \times 1 = 4$ . А если выбрать маршрутом путь 4-2-5, то расстояние равно 2, а максимальное расстояние до этого маршрута равно 3, *крутость* маршрута равна  $3 \times 2 = 6$ . *Крутость* маршрута 6-3-7 также равна 6, остальные маршруты имеют меньшую *крутость*.

В третьем тестовом примере есть только два *конечных* города 1 и 5, поэтому есть ровно один маршрут 1-2-3-4-5, расстояние равно 4, а максимальное расстояние до этого маршрута равно 0, т.к. все города лежат на этом маршруте. Соответственно *крутость* равна  $4 \times 0 = 0$ .