

Задача А. Горная трасса

Имя входного файла: `g.in`
Имя выходного файла: `g.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 megabytes

В окрестностях Алматы оборудован туристический горный круговой маршрут (старт и финиш находятся в одной точке). Будем моделировать этот маршрут n ступеньками одинаковой ширины, задавая каждую i -ю ступеньку ее высотой над уровнем моря a_i в метрах. Соседние ступеньки могут иметь одинаковую высоту. Сложность трассы определяется суммарным значением спусков и подъемов. Или формально,

$$difficulty = |a_1 - a_2| + |a_2 - a_3| + \dots + |a_{n-1} - a_n| + |a_n - a_1|.$$

Оказалось, что первоначально сконструированная трасса слишком сложна для туристов. Для того чтобы ее упростить, мы можем использовать k блоков. Ширина каждого блока совпадает с шириной ступеньки, а высота равна 1 метру. Любой из блоков можно положить на любую ступеньку или ранее положенный блок или не использовать совсем.

На сколько таким образом можно уменьшить сложность трассы в целом?

Формат входного файла

В первой строке входной файла находятся числа N ($2 \leq N \leq 10^6$) и k ($1 \leq k \leq 10^9$). В следующей строке находятся N целых неотрицательных чисел — высоты каждой из ступенек.

Формат выходного файла

Запишите в выходной файл одно число — максимально возможное уменьшение сложности трассы.

Примеры

| <code>g.in</code> | <code>g.out</code> |
|-------------------------|--------------------|
| 4 5 4 3 2 1 | 4 |
| 3 2 1 2 1 | 2 |
| 7 1000 4 3 3 2 3 2 1 | 8 |

Note

В первом примере сложность трассы составляла 6 (три спуска на 1 и подъем на 3 с учетом круговой трассы). Положив один блок на предпоследнюю ступеньку и два на последнюю, мы уменьшим сложность на 4. Этот же ответ можно было получить и с использованием всех 5 блоков. Но улучшить ответ нельзя.

Для 30% баллов $N \leq 100$, $k \leq 1000$.

Для 60% баллов $N \leq 100\,000$

Задача В. Муравьи

Имя входного файла: `e.in`
Имя выходного файла: `e.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

На координатной сетке расположено прямоугольное поле с противоположными углами в точках $(0,0)$ и (W, H) . По линиям сетки внутри поля бегают K муравьев. Каждый муравей может бежать в одном из 4 направлений и со скоростью 1 клетка в секунду.

Так как муравьи очень гордые создания, они никогда не уступают дорогу. Поэтому, если два муравья сталкиваются лоб в лоб, они мгновенно разворачиваются и продолжают идти в обратном направлении. Если два муравья, бегут перпендикулярно не обращают никакого внимания друг на друга. Если же муравей достигает границы поля, то он также разворачивается и бежит обратно.

Вам дано положение и направление движения всех муравьев в начальный момент времени. Определите их положение и направление движения через T секунд.

Формат входного файла

Первая строка входного файла содержит 4 целых числа: W, H, K, T ($1 \leq W, H, K \leq 100$, $1 \leq T \leq 10^9$).

Каждая из следующих K строк содержит по 3 целых числа: X_i, Y_i, D_i , где (X_i, Y_i) — координаты муравья, D_i — направление его движения ($0 < X_i < W$, $0 < Y_i < H$, $1 \leq D_i \leq 4$). $D_i = 1$, если муравей движется в сторону увеличения X , $D_i = 2$ — в сторону увеличения Y , $D_i = 3$ — в сторону уменьшения X , $D_i = 4$ — в сторону уменьшения Y .

Все числа в строках разделены пробелом. Все муравьи находятся в разных точках.

Формат выходного файла

Выходной файл должен содержать ровно K строк — по одной для каждого муравья в том же порядке, в котором информация о них дается во входном файле. Каждая строка должна содержать по 3 целых числа, разделенных пробелом — координаты муравья и направление его движения.

Примеры

| <code>e.in</code> | <code>e.out</code> |
|---------------------------|--------------------|
| 4 4 2 3 1 1 1 3 3 4 | 4 1 3 3 0 2 |
| 4 4 2 4 1 1 1 3 3 4 | 3 1 3 3 1 2 |
| 4 4 2 2 1 1 1 3 1 3 | 1 1 3 3 1 1 |
| 4 4 2 2 2 1 1 3 1 3 | 1 1 3 4 1 3 |

Note

Для 60% баллов $1 \leq T \leq 1000$.

Задача С. Биочипы

Имя входного файла: `d.in`
Имя выходного файла: `d.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 512 megabytes

Учённые нашли биочип, который при определённых условиях делится на несколько новых биочипов. При этом биочип-родитель прекращает свое существование. Биочипы-потомки обладают каждый своим объемом памяти. Далее биочип можно или использовать (запретив дальнейшее деление), или он будет делиться дальше по заранее известной схеме.

Учёные составили древовидные схемы деления для разных биочипов и точно знают структуру дерева из N элементов, включая объем памяти каждого из потенциальных потомков. Им нужна программа, которая выберет из этого дерева ровно M биочипов, суммарный объем памяти которых максимален. Обратите внимание, что если мы выбираем какой-то биочип, то ни один из его предков или потомков быть выбран не может.

Формат входного файла

В первой строке входного файла находятся два целых числа N и M — количество элементов дерева и количество выбираемых биочипов ($1 \leq N \leq 200\,000$, $1 \leq M \leq 500$).

Следующие N строк содержат по два неотрицательных целых числа, разделенных пробелом: номер родителя в дереве и размер собственной памяти x ($1 \leq x \leq 1000$). Каждый биочип идентифицируется числом между 1 и N включительно, строка входного файла с номером i содержит информацию о биочипе с номером $i - 1$. Ровно один биочип не имеет родителя, в качестве номера его родителя записан 0.

Входные данные таковы, что M биочипов выбрать можно.

Формат выходного файла

Выходной файл должен содержать одно целое число — максимально возможный объем памяти, выбранных M биочипов.

Примеры

| d.in | d.out |
|--------|-------|
| 7 3 | 300 |
| 2 100 | |
| 0 1000 | |
| 2 150 | |
| 3 100 | |
| 3 5 | |
| 5 100 | |
| 2 50 | |

Note

Для 20% баллов $N \leq 20$, $M \leq 10$.

Для 60% баллов $N \leq 10\,000$, $M \leq 100$.