

Задача А. Петя и прямоугольники

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 0.25 секунд |
| Ограничение по памяти: | 256 мегабайт |

Маленький Петя очень любит прямоугольники. Петя дал маме список прямоугольников, которые он хочет получить в подарок на Новый Год. Каждый прямоугольник характеризуется w и высотой h .

Мама хочет сделать Пете приятное и купить все прямоугольники из его списка. Мама отправилась в магазин и узнала, что цена одного прямоугольника равна его площади. К ее счастью, в магазине действует предновогодняя акция, позволяющая покупать прямоугольники не по одному, а сразу наборами. Стоимость одного набора равна ширине самого широкого прямоугольника, умноженной на высоту самого высокого прямоугольника из этого набора. Обратите внимание, что поворачивать прямоугольники (тем самым меняя местами ширину и высоту) нельзя. Помогите маме Пети купить все прямоугольники из списка ее сына, потратив на это наименьшее количество денег.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 2 \cdot 10^5$) — количество прямоугольников в списке Пети. В каждой из следующих n строк записаны по 2 целых положительных числа, не превышающих 10^6 , — ширина и высота очередного прямоугольника.

Формат выходных данных

Выведите одно число — наименьшее количество денег, которое может потратить мама чтобы купить Пете все прямоугольники из его списка.

Пример

| стандартный ввод | стандартный вывод |
|--------------------------------------|-------------------|
| 4 100 1 15 15 20 5 1 100 | 500 |

Задача В. Автобусные остановки

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1.5 секунд |
| Ограничение по памяти: | 256 мегабайт |

В деревне есть n домов, расположенных вдоль главной дороги, которую можно воспринимать как числовую прямую. i -й дом имеет координату x_i .

Жители предпочитают автобусные остановки рядом с их домом, и чем дальше автобусная остановка, тем более несчастливы они. *Недовольство* дома определяется как квадрат расстояния между домом и ближайшей к нему автобусной остановкой. Ваша задача — построить k автобусных остановок вдоль главной дороги так, чтобы сумма недовольств домов была минимальна.

Обратите внимание: остановка может быть построена в любой точке числовой прямой, необязательно совпадающей с точкой какого-то из домов.

Формально, пусть ближайшая остановка к i -му дому находится в точке p_i . Тогда вы хотите минимизировать:

$$\sum_{i=1}^n |x_i - p_i|^2$$

Формат входных данных

Первая строка содержит два целых числа n, k ($1 \leq n \leq 5 \cdot 10^4$, $1 \leq k \leq \min(n, 100)$).

Вторая строка содержит n целых чисел x_i ($1 \leq x_i \leq 10^5$).

Формат выходных данных

Выведите одно число — ответ с относительной или абсолютной погрешностью не более 10^{-6} .

Пример

| стандартный ввод | стандартный вывод |
|------------------|--------------------|
| 3 2 1 2 4 | 0.5000000000000000 |

Замечание

Пусть построили автобусные остановки в координатах 1.5 и 4.0. Тогда:

- Недовольство первого дома: $(x_1 - p_1)^2 = (1.0 - 1.5)^2 = 0.25$
- Недовольство второго дома: $(x_2 - p_1)^2 = (2.0 - 1.5)^2 = 0.25$
- Недовольство третьего дома: $(x_3 - p_2)^2 = (4.0 - 4.0)^2 = 0.00$

Таким образом, суммарное недовольство равно $0.25 + 0.25 + 0.00 = 0.5$

Задача С. Оптимальное бинарное дерево поиска

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

Рассмотрим множество $S = \{e_1, e_2, \dots, e_n\}$, состоящее из n различных элементов таких, что $e_1 < e_2 < \dots < e_n$. Рассмотрим бинарное дерево поиска, состоящее из элементов S . Чем чаще производится запрос к элементу, тем ближе он должен располагаться к корню. Стоимостью $cost$ доступа к элементу e_i из S в дереве будем называть значение $cost(e_i)$, равное числу ребер на пути, который соединяет корень с вершиной, содержащей элемент. Имея частоту запросов к элементам из $S - f(e_1), f(e_2), \dots, f(e_n)$ — определим общую стоимость дерева следующим образом:

$$\sum_{i=1}^n f(e_i) \cdot cost(e_i)$$

Дерево, имеющее наименьшую стоимость, считается наилучшим для поиска элементов из S . Именно поэтому оно называется Оптимальным Бинарным Деревом Поиска. Ваша задача — найти стоимость Оптимального Бинарного Дерева Поиска.

Формат входных данных

Состоит из нескольких тестов, каждый из которых расположен в отдельной строке. Первое число в строке n ($1 \leq n \leq 5000$) указывает на размер множества S . Следующие n неотрицательных целых чисел описывают частоты запросов элементов из S : $f(e_1), f(e_2), \dots, f(e_n)$. Известно, что $0 \leq f(e_i) \leq 100$. Сумма n по всем тестам не больше 5000.

Формат выходных данных

Для каждого теста в отдельной строке выведите стоимость Оптимального Бинарного Дерева Поиска.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1 5 | 0 |
| 3 10 10 10 | 20 |
| 3 5 10 20 | 20 |
| 6 1 3 5 10 20 30 | 63 |

Задача D. Доставка пиццы

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 512 мегабайт |

Флатландия — одномерная страна. Это значит, что каждая точка имеет только одну координату. Во Флатландии все любят пиццу (потому что она достаточно плоская).

Там есть n пиццерий и m покупателей. i -я пиццерия находится в точке s_i , а i -й покупатель — в точке c_i . Координаты любых двух пиццерий различны, но координаты покупателей могут совпадать.

Каждый покупатель хочет заказать пиццу и потратить минимально возможное количество денег. i -я пиццерия продаёт пиццу по цене p_i . Доставка из точки x_1 в точку x_2 стоит $(x_1 - x_2)^2$.

К сожалению, некоторые покупатели не любят некоторые пиццерии, поэтому они не будут заказывать пиццу оттуда. А именно, i -й покупатель не закажет пиццу из пиццерий $d_{i,1}, \dots, d_{i,k_i}$.

Для каждого покупателя найдите цену, за которую он закажет пиццу.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 200000$) — количество пиццерий и покупателей, соответственно.

i -я из следующих n строк содержит два целых числа s_i и p_i ($0 \leq s_i \leq 10^9$, $1 \leq p_i \leq 10^9$) — координата i -й пиццерии и цена пиццы там, соответственно.

i -я из следующих m строк содержит целые числа c_i , k_i , $d_{i,1}, \dots, d_{i,k_i}$ ($0 \leq c_i \leq 10^9$, $0 \leq k_i \leq n - 1$, $1 \leq d_{i,j} \leq n$) — координата i -го покупателя, количество пиццерий, которые он не любит, и номера этих пиццерий, соответственно.

Также гарантируется, что $\sum k_i \leq 400000$.

Формат выходных данных

Выведите m чисел, i -е из которых — цена, за которую i -й покупатель закажет пиццу.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 3 | 11 |
| 1 7 | 34 |
| 10 5 | 13 |
| 8 9 | |
| 3 0 | |
| 3 1 1 | |
| 6 2 1 2 | |

Замечание

Первый покупатель любит все пиццерии, поэтому закажет пиццу из первой. Это будет стоить $7 + (3 - 1)^2 = 11$.

Второй покупатель не любит пиццу из первой пиццерии, несмотря на то, что это самый дешёвый вариант. Он закажет пиццу из второй. Это будет стоить $9 + (10 - 3)^2 = 34$.

Третий покупатель любит пиццу только из третьей пиццерии, поэтому он закажет её там, и это будет стоить $9 + (8 - 6)^2 = 13$.

Задача E. Транспортировка кошек

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

Zxc960115 содержит большое хозяйство. Он кормит m милых кошечек и держит у себя p кормильщиков. Через ферму проходит прямая дорога, а вдоль дороги расположено n холмов, пронумерованных от 1 до n , слева направо. Расстояние от холма i до $i-1$ равняется d_i метров. Кормильщики живут на холме 1.

Однажды кошечкам захотелось порезвиться и они разбежались. Кошка i пошла к холму h_i , дошла до него в момент времени t_i , а затем стала ждать кормильщика на холме h_i . Кормильщики должны собрать всех разбежавшихся кошек. Каждый кормильщик идет прямо от холма номер 1 до холма номер n , не останавливаясь у какого-либо холма, и собирает всех кошек, **ожидающих** на каждом холме. Кормильщики двигаются со скоростью 1 в единицу времени и достаточно сильны, чтобы собрать сколько угодно кошек.

Например, пусть имеется два холма ($d_2 = 1$) и одна кошечка, которая дошла до холма 2 ($h_1 = 2$) в момент времени 3. Тогда, если кормильщик отправится за кошками от холма 1 в момент времени 2 или 3, то он сможет забрать эту кошку. Но если он отправится от холма 1 в момент времени 1, то он не сможет этого сделать. Если кормильщик отправится за кошкой в момент времени 2, то кошка будет ждать его 0 единиц времени, если же он отправится в момент времени 3, то кошка будет ждать его 1 единицу времени.

Ваша задача — составить расписание отправки от холма 1 для кормильщиков так, чтобы общее время ожидания кошек до того как их заберут было минимальным.

Формат входных данных

В первой строке входных данных содержится три целых числа n, m, p ($2 \leq n \leq 10^5, 1 \leq m \leq 10^5, 1 \leq p \leq 100$).

Во второй строке содержится $n - 1$ положительных целых чисел d_2, d_3, \dots, d_n ($1 \leq d_i < 10^4$).

В каждой из следующих m строк содержится по два целых числа h_i и t_i ($1 \leq h_i \leq n, 0 \leq t_i \leq 10^9$).

Формат выходных данных

Выведите целое число, минимальную сумму времен ожидания всех кошек.

Пример

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 4 6 2 1 3 5 1 0 2 1 4 9 1 10 2 10 3 12 | 3 |

Задача F. Сбежать через лист

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 0.3 секунд |
| Ограничение по памяти: | 256 мегабайт |

Дано дерево на n вершинах (пронумерованных от 1 до n) с корнем в вершине 1. В вершине i записаны два числа: a_i и b_i .

Вы можете прыгнуть из вершины в любую вершину в её поддереве. Стоимость такого прыжка из вершины x в вершину y равна произведению a_x и b_y . Суммарная стоимость пути между вершинами, состоящего из нескольких прыжков равна сумме стоимостей прыжков в нём. Для каждой вершины посчитайте минимальную стоимость пути от неё до какого-либо листа. Обратите внимание, что корень дерева не является листом, даже если имеет степень 1.

Учтите, что нельзя совершать прыжок из вершины в ту же вершину.

Формат входных данных

В первой строке содержится целое число n ($2 \leq n \leq 10^5$) — количество вершин в дереве.

Во второй строке через пробел заданы n целых чисел a_1, a_2, \dots, a_n ($-10^5 \leq a_i \leq 10^5$).

Во третьей строке через пробел заданы n целых чисел b_1, b_2, \dots, b_n ($-10^5 \leq b_i \leq 10^5$).

В следующих $n - 1$ строках содержатся пары целых чисел u_i и v_i ($1 \leq u_i, v_i \leq n$), разделённых пробелом, обозначающие ребро между вершинами u_i и v_i в дереве.

Формат выходных данных

Выведите n целых чисел через пробел, i -е из которых обозначает минимальную стоимость, чтобы добраться от вершины с номером i до какого-либо листа.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 3 2 10 -1 7 -7 5 2 3 2 1 | 10 50 0 |
| 4 5 -10 5 7 -8 -80 -3 -10 2 1 2 4 1 3 | -300 100 0 0 |

Замечание

В первом тестовом примере вершина 3 сама является листом, поэтому ответ равен 0. Для вершины 2 прыжок в вершину 3 стоит $a_2 \times b_3 = 50$. Для вершины 1 прыжок в вершину 3 стоит $a_1 \times b_3 = 10$.

Во втором тестовом примере вершины 3 и 4 являются листьями, поэтому ответ для них равен 0. Для вершины 2 прыжок в вершину 4 стоит $a_2 \times b_4 = 100$. Для вершины 1 необходимо сначала прыгнуть в вершину 2 прыжком стоимостью $a_1 \times b_2 = -400$, а затем прыгнуть из 2 в 4 за $a_2 \times b_4 = 100$.

Задача G. Очередная задача минимизации

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Дан массив из n чисел $a_1 \dots a_n$. Стоимостью подотрезка элементов в массиве назовем количество неупорядоченных пар различных позиций внутри подотрезка, содержащих одинаковые элементы. Разбейте массив на k непересекающихся непустых подотрезков таких, что сумма их стоимостей минимальна. Каждый элемент массива должен попасть ровно в один подотрезок.

Формат входных данных

Первая строка содержит два целых числа n и k ($2 \leq n \leq 10^5$, $2 \leq k \leq \min(n, 20)$) — размер массива и количество отрезков, на которые надо его разбить.

Следующая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — элементы массива.

Формат выходных данных

Выведите одно число — минимальную стоимость разбиения массива на подотрезки.

Примеры

| стандартный ввод | стандартный вывод |
|-----------------------------------|-------------------|
| 7 3 1 1 3 3 3 2 1 | 1 |
| 10 2 1 2 1 2 1 2 1 2 1 2 | 8 |
| 13 3 1 2 2 2 1 2 1 1 1 2 2 1 1 | 9 |

Замечание

В первом примере оптимально разбить последовательность на три подпоследовательности: $[1]$, $[1, 3]$, $[3, 3, 2, 1]$. Стоимости равны 0, 0 и 1, поэтому ответ равен 1.

Во втором примере оптимально разбить подпоследовательность на две половины. Стоимость каждой половины равна 4.

В третьем примере оптимально разбить следующим образом: $[1, 2, 2, 2, 1]$, $[2, 1, 1, 1, 2]$, $[2, 1, 1]$. Стоимости равны 4, 4, 1.