

Задача F. Безопасная дорога

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

Панкрат играет в игру «Панкрафт» — копию популярной игры, где всё сделано из кубических блоков. Он уже построил дом и выкопал шахту. Между домом и шахтой проложена прямая дорога, состоящая из n блоков.

Пока Панкрат идёт по этой дороге, на него могут напасть монстры. Панкрату это не нравится, и он решил обезопасить свой путь до шахты. Панкрат знает, что монстры боятся света, поэтому он хочет установить вдоль дороги факелы. Каждый факел освещает k блоков влево и k блоков вправо (включая блок, на котором он установлен). То есть факел, установленный на блоке i , освещает все блоки с номерами от $\max(1, i - k)$ до $\min(n, i + k)$.

Однако в этой версии игры игрок не может создавать и расставлять факелы самостоятельно. Зато это может сделать внутриигровой персонаж — странствующий торговец. Странствующий торговец берёт a_i изумрудов за установку факела на блоке с индексом i .

Панкрат хочет осветить всю дорогу, чтобы на него не нападали монстры. Помогите ему найти способ сделать это с минимальными затратами изумрудов. Считайте, что у Панкрата бесконечное количество изумрудов, а у торговца бесконечное количество факелов.

Формат входных данных

Первая строка содержит два целых числа n и k — соответственно, количество блоков на дороге и радиус освещения одного факела ($1 \leq n \leq 5 \cdot 10^6$, $0 \leq k \leq 5 \cdot 10^6$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$, где a_i — стоимость установки факела на блоке с индексом i ($0 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — наименьшую цену за освещение всей дороги.

Система оценки

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

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$1 \leq n \leq 20$, $0 \leq k \leq 20$ $0 \leq a_i \leq 10^9$	15	—	первая ошибка
2	$1 \leq n \leq 1000$, $0 \leq k \leq 1000$ $0 \leq a_i \leq 10^9$	50	—	первая ошибка
3	$1 \leq n \leq 10^5$, $0 \leq k \leq 10^5$ $0 \leq a_i \leq 10^9$	85	—	первая ошибка
4	Нет дополнительных ограничений	100	1, 2, 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
7 1 0 3 1 2 4 2 3	3
10 3 2 3 5 10 23 4 3 2 1 4	4

Замечание

Если k равен нулю, то факелы освещают только блок, на котором они стоят.

Задача G. Заполнение хордами

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Артему на день рождения друзья подарили окружность, на которой отмечены n точек, где все расстояния между соседними точками одинаковые. Точки пронумерованы от 1 до n по часовой стрелке. Также на окружности уже нарисованы k хорд, при этом никакие две хорды не пересекаются. Две хорды пересекаются, если у них есть хотя бы одна общая точка (включая концы хорд).

Перед тем как гордо повесить свою окружность на стену, Артему нужно добавить в нее наибольшее возможное количество хорд так, чтобы указанное выше свойство все еще выполнялось. Так как Артем уже очень устал от программирования, он хочет, чтобы вы сделали это за него. Если есть несколько способов добавить наибольшее количество хорд, то вы можете сделать это любым из них.

Формат входных данных

В первой строке даны два целых числа — n и k ($1 < n \leq 10^5$, $0 \leq k \leq \lfloor \frac{n}{2} \rfloor$).

В следующих k строках описаны уже нарисованные хорды: в i -й строке записаны два числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — номера точек, соединенных i -ой хордой.

Гарантируется, что никакие две хорды не пересекаются.

Формат выходных данных

В первую строку выведите одно число m — наибольшее количество хорд, который можно добавить в окружность.

В следующих m строках выведите m хорд в таком же формате, как и во входных данных.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

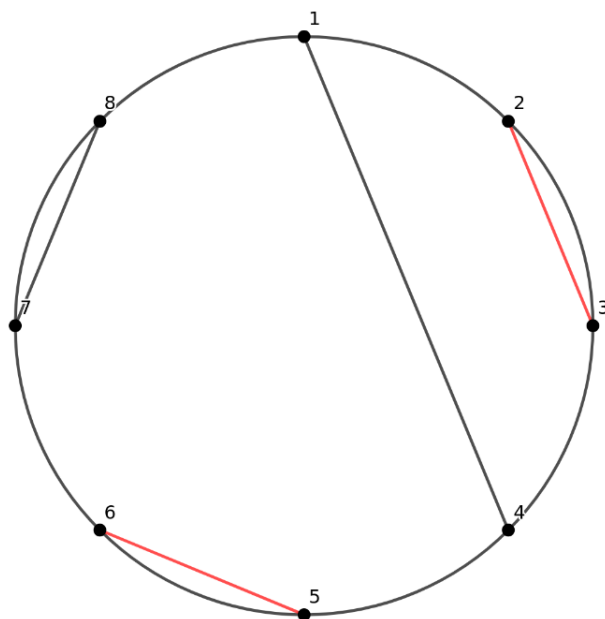
№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$k = 0$	5	—	первая ошибка
2	$k = 1$	10	У, 1	первая ошибка
3	$k = 2$	20	У, 1	первая ошибка
4	нет дополнительных ограничений	65	У, 1	первая ошибка

Примеры

стандартный ввод	стандартный вывод
8 2 1 4 7 8	2 2 3 5 6
19 3 7 3 15 8 9 11	5 4 5 12 13 1 2 16 17 18 19
10 3 1 3 5 7 8 10	0

Замечание

Иллюстрация к примеру 1:



Красным цветом выделены хорды, которые Артем добавит к окружности.

Задача Н. Погоня

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

Скоро состоится международная олимпиада школьников «Туймаада» 2026, Алёша является одним из разработчиков задач. В процессе разработки очередной задачи-игры он не смог придумать к ней решение, поэтому попросил помощи у своего друга Тимура. Тот, в свою очередь, смог справиться с данной задачей, но решил позаимствовать её для другой, тоже международной олимпиады школьников «Адаамйут». Алёша должен остановить его, пока не слишком поздно! Друзья решили смоделировать эту игру и определить владельца задачи в честной борьбе.

По условиям игры, имеется n городов и m авиарейсов, каждый из которых позволяет полететь из города u_i в город v_i за w_i рублей. Авиарейсы однонаправленные, то есть с помощью рейса i нельзя полететь из города v_i в город u_i . В один и тот же день каждый из игроков выполняет не более одного перелета. В первый день Алёша находится в городе a , а Тимур — в городе t_1 . День под номером j ($1 \leq j < k$) можно описать следующим алгоритмом:

1. Алёша либо ничего не делает, либо, если он находится в городе u_i , с помощью рейса i может полететь в город v_i , заплатив при этом за перелёт w_i рублей.
2. Тимур летит из города t_j в город t_{j+1} (гарантируется, что существует такой i , что $u_i = t_j$ и $v_i = t_{j+1}$). Заметьте, что Алёша знает путь Тимура заранее.
3. Если Алёша и Тимур находятся в одном и том же городе, то Алёша ловит Тимура и погоня заканчивается (Алёша становится владельцем задачи).

Если Алёша не смог поймать Тимура за $k-1$ дней, то Тимур скрывается и становится владельцем задачи.

Очевидно, что способов поймать Тимура может быть несколько, поэтому Алёша просит вас найти минимальную стоимость поимки Тимура.

Формат входных данных

Первая строка содержит четыре целых числа n , m , a и k — количество городов и авиарейсов, начальный город Алёши и количество городов на пути у Тимура соответственно ($2 \leq n \leq 150$; $1 \leq m \leq 300$; $1 \leq a \leq n$; $1 \leq k \leq 10^5$).

Вторая строка содержит k целых чисел $t_1, t_2, \dots, t_k$ — города на пути у Тимура ($1 \leq t_i \leq n$; $t_1 \neq a$).

Следующие m строк содержат по три целых числа u_i , v_i и w_i — авиарейс из города u_i в город v_i , который стоит w_i рублей ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; нет таких i и j ($i \neq j$, что $u_i = u_j$ и $v_i = v_j$; $0 \leq w_i \leq 10^6$).

Формат выходных данных

Выведите минимальную стоимость поимки Тимура или -1 , если Тимура не удастся поймать.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$m = n - 1, v_i = u_i + 1$	10	—	первая ошибка
2	$m = (n - 1) \cdot 2$; для каждого i есть такой j , что $u_i = v_j, v_i = u_j$; из произвольной вершины можно добраться до любой вершины	15	1	первая ошибка
3	$n \leq 9$	15	—	первая ошибка
4	Нет дополнительных ограничений	60	У, 1, 2	первая ошибка

Примеры

стандартный ввод	стандартный вывод
7 7 1 4 4 5 6 7 1 2 2 2 3 1 2 6 3 3 6 1 4 5 1 5 6 1 6 7 1	5
7 7 1 4 4 5 6 7 1 2 2 2 3 1 2 7 3 3 7 1 4 5 1 5 6 1 6 7 1	4
7 5 1 4 4 5 6 7 1 2 2 2 3 1 4 5 1 5 6 1 6 7 1	-1

Замечание

В первом примере можно заметить, что если Алёша не сможет за максимум 2 дня достичь города 6, то он в любом случае проигрывает. Таким образом, оптимальным для Алёши является путь $1 \rightarrow 2 \rightarrow 6$, стоимость которого равна 5 рублям.

Второй пример похож на первый, за исключением того, что теперь лимит дней равен 3. Поэтому в данном примере можно выбрать более длинный, но дешёвый путь $1 \rightarrow 2 \rightarrow 3 \rightarrow 7$, который стоит 4 рубля.

В третьем примере Алёша не сможет поймать Тимура.

Задача I. Заработок

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

Дима — начинающий предприниматель, мечтающий организовать лучший туристический маршрут по Якутии. Он уже составил карту, включающую N городов, соединённых двусторонними дорогами. Карта представляет собой дерево: между каждой парой городов существует единственный путь.

Каждая дорога характеризуется двумя целыми числами:

- A_i — потенциальный доход, который можно получить, если включить эту дорогу в маршрут;
- B_i — время, необходимое для прохождения этой дороги.

Некоторые города имеют аэропорты, и маршрут **должен** начинаться и заканчиваться в таких городах.

Дима хочет выбрать непрерывный путь $(i_1, i_2, \dots, i_K)$ по дорогам, начинающийся и заканчивающийся в городах с аэропортами, чтобы максимизировать доходность маршрута, определяемую как:

$$\frac{A_{i_1} + A_{i_2} + \dots + A_{i_K}}{B_{i_1} + B_{i_2} + \dots + B_{i_K}}.$$

Ваша задача — найти дорогу с максимально возможной доходностью маршрута удовлетворяющего вышеуказанным условиям.

Формат входных данных

В первой строке содержатся два целых числа N и M — количество городов и городов, в которых есть аэропорт ($2 \leq M \leq N \leq 10^5$).

Во второй строке содержатся M чисел $T_1, T_2, \dots, T_M$ — номера городов, в которых есть аэропорты ($1 \leq T_i \leq N$). Гарантируется, что все T_i попарно различны.

Далее следуют $N - 1$ строк, описывающих дороги. Каждая строка содержит 4 целых числа u_i, v_i, A_i, B_i , означающих, что существует дорога между городами u_i и v_i с доходом A_i и временем прохождения B_i ($1 \leq u_i, v_i \leq N, 1 \leq A_i, B_i \leq 10^9$). Гарантируется, что данные дороги образуют дерево.

Формат выходных данных

В первой строке выведите одно число K ($2 \leq K \leq N$) — число вершин в найденном пути.

Во второй строке выведите K чисел i_j — номера вершин в пути. Вершины $i_1, i_2, \dots, i_K$ должны образовывать простой путь длины хотя бы 1. Если существует несколько возможных ответов — выведите любой.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$u_i = i, v_i = i + 1, N \leq 500$	5	—	первая ошибка
2	$u_i = i, v_i = i + 1, N \leq 5000$	5	1	первая ошибка
3	$u_i = i, v_i = i + 1$	10	1, 2	первая ошибка
4	$M = 2$	5	—	первая ошибка
5	$M = N$	5	—	первая ошибка
6	$N \leq 500$	10	У, 1	первая ошибка
7	$N \leq 5000$	20	У, 1, 2, 6	первая ошибка
8	$N \leq 50000$	30	У, 1, 2, 6, 7	первая ошибка
9	Нет дополнительных ограничений	10	У, 1, 2, 3, 4, 5, 6, 7, 8	первая ошибка

Пример

стандартный ввод	стандартный вывод
6 4 2 3 5 6 2 3 5 3 3 4 1 5 1 5 12 1 6 4 13 4 5 4 9 4	3 5 4 6

Задача J. Побег из инопланетной тюрьмы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	9 секунд
Ограничение по памяти:	256 мегабайт

Талбан — бедный студент Северо-Восточного федерального университета, которого только что похитили инопланетяне. Теперь ему надо незаметно сбежать из места заточения, пока над ним не провели бесчеловечные эксперименты. В его камере есть вентиляционный люк, ведущий в комнату со спасательной капсулой. Эту комнату можно считать бесконечной двумерной плоскостью, на которой находится множество вышек.

Система безопасности у инопланетян очень странная — до постройки этой тюрьмы у них был массив из n векторов (Vx_i, Vy_i) . Сначала инопланетяне выбрали координаты первой вышки (Sx, Sy) и какой-то подотрезок этого массива $[L, R]$. Затем для каждого числа i от L до R все существующие вышки в точке (x, y) «клонировали» сами себя на координаты $(x + Vx_i, y + Vy_i)$. То есть количество вышек увеличивалось вдвое после каждой операции. Эти вышки работают следующим образом — если человек пройдёт строго насквозь через какой-то отрезок, образованный 2 вышками, или попадёт строго внутрь какого-то треугольника, образованного 3 вышками, то он мгновенно испепелится на месте. Для лучшего понимания условия посмотрите пояснения к примерам.

К сожалению, Талбан пока толком не знает ни координаты изначальной вышки, ни какой подотрезок массива инопланетяне выбрали для клонирования. Более того, он даже не знает координаты вентиляции и спасательной капсулы! Но к его счастью, Дайаану, ясновидящую подругу Талбана, тоже похитили и поместили в камеру рядом с ним. Она предсказала q вариантов развития событий — i -й вариант состоит из 8 чисел: $L_i, R_i, Sx_i, Sy_i, Ax_i, Ay_i, Bx_i, By_i$. Он означает, что инопланетяне выбрали координаты первой вышки (Sx_i, Sy_i) и подотрезок $[L_i, R_i]$ для их клонирования, а также что вентиляционный люк и спасательная капсула находятся в точках (Ax_i, Ay_i) и (Bx_i, By_i) соответственно.

Так как у Талбана и Дайааны осталось совсем немного времени, им нужно для каждого из q возможных событий вычислить минимальное расстояние, которое они должны пройти от вентиляции до спасательной капсулы, чтобы не испепелиться по пути, или сказать, что в этом случае сбежать невозможно.

Формат входных данных

В первой строке заданы три целых числа n, q и G — количество векторов в массиве инопланетян, количество вариантов событий Дайааны и номер подзадачи соответственно ($1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$, $0 \leq G \leq 9$, где $G = 0$ означает тесты из условия).

Каждая из следующих n строк содержит по два целых числа Vx_i и Vy_i — координаты вектора i ($|Vx_i|, |Vy_i| \leq 10^4$). Гарантируется, что все векторы ненулевые.

Затем каждая из следующих q строк содержит по восемь целых чисел $L_i, R_i, Sx_i, Sy_i, Ax_i, Ay_i, Bx_i, By_i$ — границы отрезка клонирования, координаты первой вышки, координаты вентиляции и координаты спасательной капсулы соответственно ($1 \leq L_i \leq R_i \leq n$, $|Sx_i|, |Sy_i| \leq 10^6$, $|Ax_i|, |Ay_i|, |Bx_i|, |By_i| \leq 10^9$).

Формат выходных данных

Для каждого из q вариантов событий выведите единственное вещественное число — минимальное возможное расстояние, которое Талбан и Дайаана должны пройти от вентиляции до спасательной капсулы, или -1 , если они в любом случае испепелятся. Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не превосходит 10^{-4} .

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены. Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач дополнительно указана буква У.

№	Дополнительные ограничения	Баллы	Необходимые подзадачи	Информация о проверке
1	$Vx_i, Vy_i \leq 0$ для всех $1 \leq i \leq n$; $Sx_i, Sy_i \leq 0$ и $Ax_i, Ay_i, Bx_i, By_i \geq 0$ для всех $1 \leq i \leq q$	7	—	первая ошибка
2	$L_i = R_i$ для всех $1 \leq i \leq q$	12	—	первая ошибка
3	$R_i - L_i \leq 1$ для всех $1 \leq i \leq q$	12	2	первая ошибка
4	$L_i = 1, R_i = n, Ax_i = Bx_i$ и $Ay_i = By_i$ для всех $1 \leq i \leq q$	13	—	первая ошибка
5	$L_i = 1, R_i = n$ для всех $1 \leq i \leq q$	13	4	первая ошибка
6	$Ax_i = Bx_i$ и $Ay_i = By_i$ для всех $1 \leq i \leq q$	13	4	первая ошибка
7	$n, q \leq 1000$	10	У	первая ошибка
8	$n, q \leq 5 \cdot 10^4$	10	У, 7	первая ошибка
9	Нет дополнительных ограничений	10	У, 1 – 8	первая ошибка

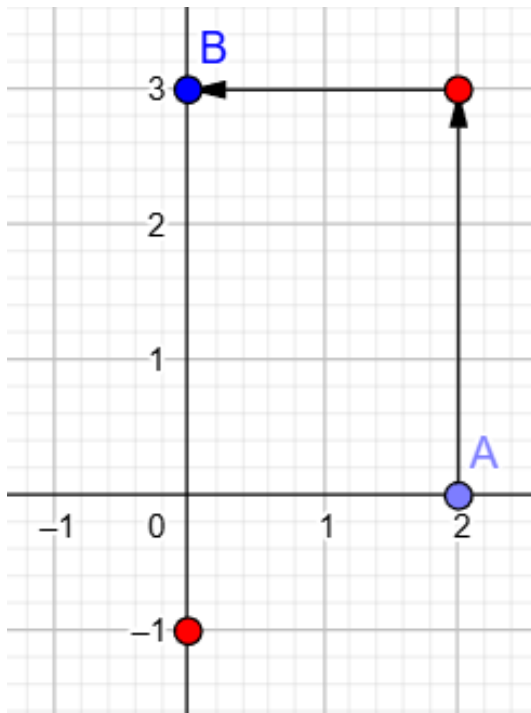
Примеры

стандартный ввод	стандартный вывод
3 5 0 2 4 1 -1 0 2 1 1 0 -1 2 0 0 3 1 3 0 0 2 0 2 8 3 3 3 1 3 2 3 2 1 3 0 0 3 1 1 2 1 3 0 0 1 2 1 2	5.0000000 8.3245553 0.0000000 -1 -1
4 3 0 -2 -2 4 4 -2 0 -2 1 1 2 3 -2 5 0 7 -2 2 2 1 -1 6 4 0 -2 1 4 0 0 -5 1 -4 4	2.8284271 8.4852814 3.1622777

Замечание

Считается, что вы прошли строго насквозь отрезок, если вы вошли внутрь отрезка с одной полуплоскости, а вышли с другой, при этом коснувшись любой точки, лежащей строго внутри отрезка. Прямая, образованная этим отрезком, не входит ни в какую полуплоскость.

В первом запросе первого примера оптимальный путь выглядит следующим образом:



Красными точками обозначены вышки. Так как мы не можем проходить строго насквозь отрезок, его надо обойти. Заметьте, что Талбан и Дайаана могут безопасно пройти через точку, содержащую вышку, так как ни одно условие испепеления не выполняется.

Во втором запросе первого примера оптимальный путь выглядит следующим образом:

