

Задача А. Стрельба из пушки

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

Галактическая Федерация хочет разрушить дом Рика при помощи новейшего планетарного уничтожителя NX5. Он представляет из себя пушку, расположенную на земле в точке $(0, 0)$, из которой можно стрелять энергетическим пучком под любым углом относительно оси X на расстояние k .

Если по пути пучок касается верха какого-то препятствия, то это препятствие уничтожается. Если же пучок сталкивается с ним, его вся энергия пучка уходит в это препятствие, уничтожает его, и пучок прекращает свое движение. Если, пролетев расстояние k , пучок ни с чем не сталкивается, то вся его энергия уходит вниз перпендикулярно поверхности земли (как молния), уничтожая препятствие на пути к земле, если оно есть. Также NX5 может выстрелить параллельно оси X , и пучок будет лететь, пока не пролетит расстояние k или не наткнется на препятствие.

На расстоянии d от устройства стоит энергетический щит высоты w , который является препятствием для пучка. На расстоянии $2d$ от земли находится дом семьи Смитов, который имеет высоту h . Найдите минимальный угол, под которым враги могут выстрелить из уничтожителя и разрушить дом, или выведите -1 , если такого угла не существует.

Формат входных данных

В одной строке и единственной строке входных данных через пробел дано четыре целых числа d, h, w, k ($1 \leq d, h, k \leq 10^9$; $0 \leq w \leq 10^9$) — расстояние от пушки до щита по координате X , высота дома, высота электрического щита и максимальная дальность полета пучка.

Формат выходных данных

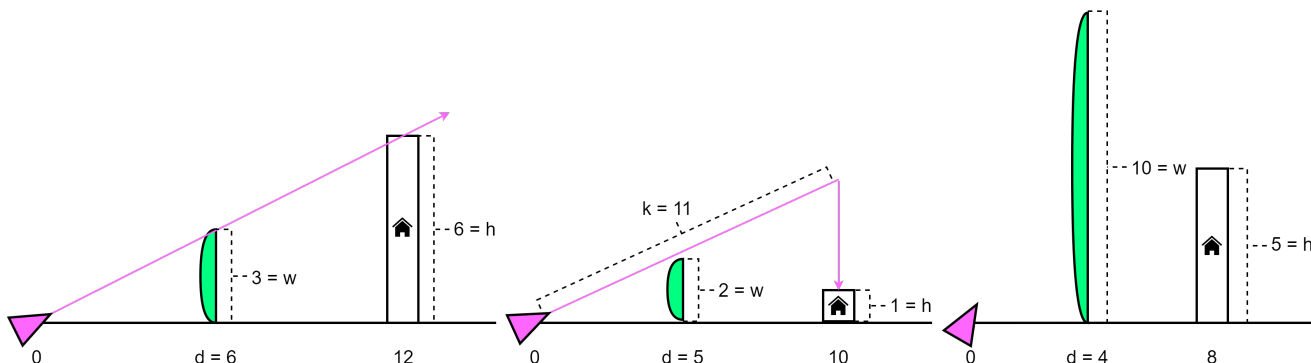
Выведите минимальный угол, под которым нужно стрелять, чтобы разрушить дом героев (в градусах с точностью до четырех знаков после запятой), или -1 , если это невозможно.

Примеры

стандартный ввод	стандартный вывод
6 6 3 20	26.5651
5 1 2 11	24.62
4 5 10 12	-1

Замечание

Ниже приведены иллюстрации к тестовым примерам.



Задача В. Иерархия цитадели

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

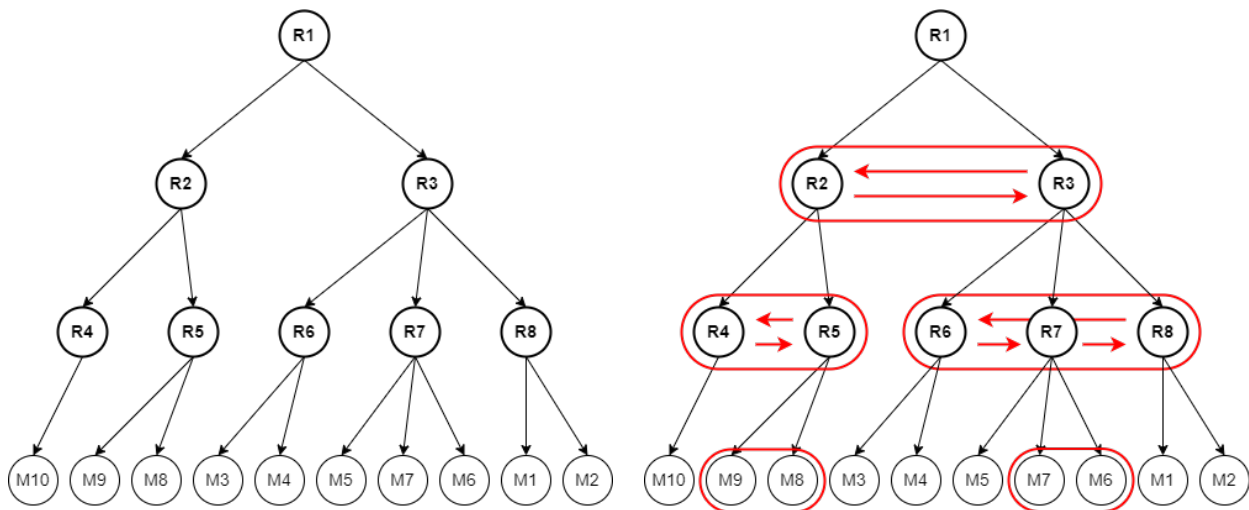
Как известно, в Цитадели Риков обитает бесчисленное множество Риков и Морти (а именно — n Риков и m Морти). Чтобы в новой Цитадели не было полного хаоса, было решено построить четкую иерархию, благодаря которой всегда можно будет быстро определить ~~какой Морти~~ кто виноват в том или ином происшествии.

Для начала было решено пронумеровать всех Риков по уменьшению важности от 1 до n , а Морти — по увеличению неважности от 1 до m . Иерархию обитателей цитадели решили изобразить в виде подвешенного дерева, при чем

- в дереве ровно n внутренних вершин и m листьев, и все листья дерева находятся на одной глубине;
- все внутренние вершины заняты Риками, а все листья заняты Морти (разумеется, все Морти должны находиться в самом низу иерархии);
- номера всех Риков на любом уровне меньше, чем номера Риков на следующих уровнях (в частности, в корне дерева всегда находится Верховный Рик под номером 1);
- на каждом уровне Рики пронумерованы по возрастанию слева-направо;
- у каждого Рика до **предпоследнего слоя** есть хотя бы два непосредственных подчиненных.

Рики, разумеется, быстро справились с построением такой иерархии, а вот Морти в панике выстроились на нижнем уровне в каком-то случайном порядке. Посмотрев на этот хаос, Рики решили, что раз они все равно не очень любят правила, то правило про упорядоченность Риков на каждом уровне можно отменить, а вот Морти на нижнем уровне надо упорядочить по возрастанию номеров слева-направо.

Для этого каждому Рику было разрешено поменять своих непосредственных подчиненных местами произвольным образом. При этом менять множество своих подчиненных, то есть брать новых или отдавать старых кому-то еще, а также перемещаться на другой уровень иерархии, запрещено. Получится ли у Риков таким образом упорядочить всех Морти по возрастанию? Ниже приведен пример возможного решения:



Изначальная иерархия и действия, необходимые для упорядочивания Морти.

Формат входных данных

В первой строке через пробел перечислены два целых числа n и m — количество Риков и Морти в Цитадели, соответственно ($2 \leq n, m \leq 10^5$).

Во второй строке через пробел перечислены m различных целых чисел a_i — номера Морти в порядке их следования в изначально построенной иерархии слева-направо ($1 \leq a_i \leq m$).

В следующей строке через пробел перечислены числа p_2, \dots, p_n — номера непосредственных начальников Риков с номерами от 2 до n ($1 \leq p_i < i$).

В следующей строке, аналогично, перечислены m целых чисел q_1, \dots, q_m — номера непосредственных начальников всех Морти, в порядке, в котором они следуют в исходной иерархии ($1 \leq q_i \leq n$). **Обратите внимание**, что q_1 — номер начальника Морти с номером a_1 , а не с номером 1.

Гарантируется, что структура иерархии соответствует заданным в условии ограничениям.

Формат выходных данных

В единственной строке выведите «YES» (без кавычек), если Рики, меняя местами непосредственных подчиненных, могут упорядочить всех Морти по возрастанию номеров, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
4 4 1 3 2 4 1 1 1 2 2 3 4	NO
8 10 10 9 8 3 4 5 7 6 1 2 1 1 2 2 3 3 3 4 5 5 6 6 7 7 7 8 8	YES

Задача С. Планеты двух измерений

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

Рик и Морти запланировали путешествие по планетам двух недавно открытых измерений. Измерениям пока не придумали названия, поэтому будем именовать их X и Y . Известно, что в измерении X всего n планет, а в Y — m .

Рик, разумеется, против того, чтобы Морти диктовал ему, как будет устроено очередное их приключение, но раз уж у Морти скоро день рождения, Рик готов один раз ему уступить.

Услышав, что Рик позволит ему полностью выбирать, какие планеты и в каком порядке они посетят, Морти заявил, что не хочет посещать одну и ту же планету несколько раз, иначе будет скучно. Также он не хочет посещать две планеты из одного и того же измерения подряд, чтобы разнообразить получаемые впечатления. При этом, начать и закончить путешествие можно на любой планете любого измерения.

Рик решил не тратить время на то, чтобы объяснить Морти, что такие условия не всегда позволяют посетить все $n + m$ планет. В конце концов, за время, сэкономленное на объяснении этого, Рик успеет после и сам посетить оставшиеся планеты.

Тем не менее, сейчас уже стоит составлять детальный план путешествия, поэтому, по данным n и m найдите максимальное количество планет, которое можно посетить, не нарушив поставленные Морти условия.

Формат входных данных

В первой строке ввода через пробел даны два целых числа n и m — количество планет в измерениях X и Y соответственно ($1 \leq n, m \leq 10^9$).

Формат выходных данных

В единственной строке выведите ответ на вопрос задачи.

Примеры

стандартный ввод	стандартный вывод
2 3	5
4 4	8
5 11	11

Задача D. Эффективный двигатель

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

Как известно, тарелка Рика летает, используя энергию, вырабатываемую обитателями карманной вселенной, созданной самим Риком. После прошлого инцидента Рик долго размышлял, какие меры предосторожности принять, чтобы снова не оказаться без возможности завести летающую тарелку в самый неподходящий момент, и наконец придумал следующее решение.

Рик решил, что теперь под капотом тарелки будет находиться не одна карманная вселенная, а целых $n!$ При чем, чтобы жители этих карманных вселенных не успевали развивать свои цивилизации до такого уровня, на котором они смогут сами изобрести свои карманные вселенные, некоторые из них Рик периодически будет перезапускать.

Изначально все n карманных вселенных находятся в замороженном состоянии и не функционируют. Механизм перезапуска устроен следующим образом: перед i -м полетом летающей тарелки Рик по очереди пройдет по всем вселенным, номера которых делятся на i (включая саму i -ю), и поменяет их состояние:

- если соответствующая вселенная была заморожена или была раньше перезапущена, Рик спустится в нее и представит ее обитателям технологию, которая позволит им вырабатывать для него энергию;
- если же вселенная уже функционировала, Рик отправляет ее на перезапуск, откатывая развитие цивилизации в ней на стартовую ступень.

Если в какой-то момент перед полетом оказывается, что ни одна вселенная не готова к выработке энергии, на этот случай у Рика есть одна запасная, так что беспокоиться не о чем.

Теперь Рика интересует, сколько вселенных будут функционировать после первых n полетов летающей тарелки. Помогите ему посчитать это количество. Полеты нумеруются от 1 до n , как и карманные вселенные.

Формат входных данных

В единственной строке дано число n — количество карманных вселенных, установленных в летающую тарелку ($1 \leq n \leq 10^{18}$).

Формат выходных данных

В единственной строке выведите ответ на задачу.

Пример

стандартный ввод	стандартный вывод
2	1

Задача Е. Инопланетные кальмары

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

Во время очередного путешествия Рик столкнулся с толпами инопланетных кальмаров. Кальмары достаточно умны, поэтому, попадая на кого-нибудь, обязательно выстраиваются в колонны, образуя гистограмму (набор столбцов с общей нижней границей). Например, колонны размеров 1, 3, 4 и 2 можно представить в виде:

```
  X
 XX
XXX
XXXX
```

Чтобы вернуться как можно скорее домой, Рик решил переместить их в другую вселенную. Для этого наш герой изобрел лазерный телепортер. Для использования этого прибора нужно выбрать несколько подряд идущих колонн одинаковой высоты h и число $0 < x \leq h$, то есть такое, что в каждой из выбранных колонн есть хотя бы x кальмаров. Тогда после использования телепортера первые x кальмаров в каждой из выбранных колонн перемещаются в другую вселенную (а высоты каждой из выбранных колонн уменьшаются на x).

Считайте, что кальмары перемещаются достаточно медленно, чтобы можно было воспринимать их как стоящих на месте. Соответственно, столбцы всегда будут иметь общую нижнюю границу и будут уменьшаться только сверху вниз.

Что с ними происходит дальше — загадка, но в рамках этой задачи не будем задаваться этим вопросом. Найдите минимальное количество раз, которое необходимо будет воспользоваться телепортером, чтобы переместить всех инопланетных кальмаров в другую вселенную.

Формат входных данных

В первой строке вводится одно единственное число n — количество колонн из кальмаров ($1 \leq n \leq 2 \cdot 10^5$).

В следующей строке дано n целых чисел a_i — количество кальмаров в каждой колонне ($0 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно единственное число — ответ на задачу.

Примеры

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

Задача F. Артефакты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

Рик и Морти обнаружили в другом измерении карту планеты, на которой изображены n пунктов раскопок, соединенных тропинками. Тропинка с номером i соединяет пункты с номерами u_i и v_i и имеет длину 1.

Разумеется, Рик сразу заметил, что количество тропинок равняется в точности $n - 1$, и из любого пункта можно добраться до любого другого. Иными словами, структура дорог и пунктов представляет из себя дерево, но для Морти это определение слишком сложное, поэтому Рик оставил Морти изучать теорию графов, а сам отправился исследовать это измерение.

Инопланетный информатор сообщил ему, что всего существует $k \leq 2$ видов артефактов, и в пункте номер i хранится артефакт вида a_i . Так очень удачно совпало, что у Рика очередное соревнование с одним из известных расхитителей космических гробниц, и для победы Рик нужно собрать по одному экземпляру каждого из видов артефактов.

Одной из проблем является то, что внутри этого измерения не работают никакие продвинутые технологии. Поэтому Рик может заранее создать порталы в запланированных стартовом и конечном пункте маршрута, а вот остальной маршрут придется пройти пешком. Чтобы сэкономить свое время, Рик хочет заранее выбрать стартовый пункт, конечный пункт и сам путь (не обязательно простой) так, чтобы пройденное им расстояние было минимальным, и при этом на пути он бы собрал все различные виды артефактов.

Рик, конечно, и сам может справиться с поиском такого кратчайшего пути, но, может быть, у вас есть время заняться этим, пока он собирает всю необходимую для путешествия экипировку?

Формат входных данных

В первой строке через пробел даны два целых числа n и k ($2 \leq n \leq 10^5$; $1 \leq k \leq 2$) — количество пунктов и необходимое количество артефактов.

В следующей строке через пробел даны n целых чисел a_i — виды артефактов в каждом пункте ($0 \leq a_i \leq k$). В случае, если $a_i = 0$, считается, что в вершине не хранится никакой из видов артефактов.

В следующих $n - 1$ строках даны пары целых чисел u_i и v_i , обозначающие наличие тропинки между пунктами u_i и v_i ($1 \leq u_i, v_i \leq n$). Гарантируется, что структура графа представляет из себя дерево.

Формат выходных данных

В случае, если невозможно собрать k различных видов артефактов выведите -1 , иначе сообщите минимальное расстояние, которое придется пройти, чтобы собрать все виды артефактов.

Примеры

стандартный ввод	стандартный вывод
5 2 1 1 2 2 2 1 3 2 3 3 4 4 5	1
5 1 1 0 1 0 1 1 3 2 3 3 4 4 5	0
4 2 0 1 2 1 1 2 2 4 1 3	2

Замечание

В первом примере можно стартовый портал поставить в вершину 1, конечный в вершине 3. Для сбора всех видов артефактов нужно будет пройти ровно по одному ребру.

Во втором примере можно стартовый портал поставить в любую вершину с первым артефактом. И так как всего существует один тип, то его сразу соберут.

В третьем примере можно стартовый портал поставить в вершину 2, конечный в вершине 3. Для сбора всех видов артефактов нужно будет пройти ровно 2 ребра.

Задача G. Незваные гости

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

Рик решил, что пока он работает над довольно серьезным изобретением в своей новой лаборатории, он хочет знать, кто за это время посещает Землю, ведь среди таких посетителей могут оказаться как люди из Галактической Федерации, так и более опасные личности.

Для этого Рик классифицировал всех возможных существ во Вселенной и разбил их на n групп по степени их опасности или подозрительности.

Система логирования устроена довольно просто, и в тот момент, когда кто-то с категорией опасности i прилетает за Землю, она делает запись $(i +)$, а когда кто-то с такой категорией опасности покидает планету — делает запись $(i -)$. Известно, что в момент запуска системы на планете находятся только люди, которых Рик вообще не воспринимает как угрозу, и поэтому не отнес ни к одной категории.

В какой-то момент Рик посмотрел на логи, в которых уже накопилось m записей, и обеспокоился тем, что из некоторых подозрительных категорий планету посещало достаточно много личностей. По записям в логах помогите Рикуну определить минимально возможное число различных людей существ из каждой категории, которые посещали Землю. Разумеется, никто не может прилететь на Землю два раза подряд, предварительно не улетев перед этим.

Формат входных данных

В первой строке ввода через пробел даны два целых числа n и m — количество категорий существ и количество записей в логах ($1 \leq n \leq 10^5$; $1 \leq m \leq 3 \cdot 10^5$).

В следующих m строках даны записи логирующей системы. В одной записи содержится число x_i и символ '+' — кто-то из x_i -й категории прилетел на Землю, или '-' — кто-то покинул планету ($1 \leq x_i \leq n$).

Формат выходных данных

Выведите в одной строке n целых чисел через пробел, i -е число должно быть равно минимально возможному количеству различных посетителей с i -й категорией опасности.

Примеры

стандартный ввод	стандартный вывод
2 4 1 + 1 + 2 + 1 -	2 1
3 10 1 + 1 + 2 + 2 - 1 - 2 + 3 + 3 - 1 - 2 -	2 1 1

Задача Н. Халат Рика

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

Рик решил постирать свой халат. Но у него не так много времени до очередного безумного приключения, поэтому вместо нормальной стирки он намочил s точек халата специальным раствором и теперь ждет, когда халат высохнет.

Раствор, созданный Риком, обладает подобием разума, поэтому вместо того, чтобы неэффективно испаряться, растекается по халату. Сам же халат можно представить в виде неориентированного графа на n вершинах и m ребрах. Точки, которые Рик намочил раствором — это k различных вершин графа. Также, как, наверное, можно было заметить, Халат Рика очень старый, поэтому в нем имеются t дырок (еще t различных вершин), и как раз через них раствор может стекать.

Рик точно не создал бы раствор, который действует не оптимально, ведь у него не так много времени, поэтому раствор из каждой изначальной точки течет до ближайшей дырки и весь через нее вытекает. На прохождение одного ребра ткани раствору требуется одна единица времени. По одному ребру одновременно может течь любой объем раствора.

Поскольку Рик снова занят, посчитайте за него, через какое время весь раствор вытечет из халата через дырки.

Формат входных данных

В первой строке через пробел даны целые числа n , m , s и t — количество вершин и ребер в халате, а также количество изначально намоченных точек и дырок, соответственно ($1 \leq n, m \leq 2 \cdot 10^5$; $1 \leq s, t \leq n$).

В следующей строке через пробел перечислены k различных целых чисел p_i — номера вершин, которые Рик намочил раствором ($1 \leq p_i \leq n$).

В следующей строке так же перечислены t различных целых чисел q_i — номера вершин, в которых в халате находятся дырки ($1 \leq q_i \leq n$).

Следующие m строк содержат описание ребер графа. В i -й из них через пробел даны два целых числа v_i и u_i , означающие, что в графе есть ребро между вершинами u_i и v_i ($1 \leq v_i, u_i \leq n$).

Формат выходных данных

В единственной строке выведите ответ на задачу — минимальное время, за которое все капли раствора смогут добраться до дырок в халате.

Пример

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