

## Задача А. Железнодорожная поездка

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

JOI Railways — компания, управляющая одной железной дорогой. На железной дороге JOI Railways существует  $n$  станций на прямой, пронумерованных от 1 до  $n$ . Для каждого  $i$  ( $1 \leq i \leq n-1$ ), станция  $i$  и станция  $i+1$  связаны железной дорогой.

JOI Railways имеют  $k$  типов поездов, работающих в обоих направлениях. Типы поездов пронумерованы целыми числами от 1 до  $k$  включительно. Каждая станция имеет уровень, которые является целым числом от 1 до  $k$  включительно. Для каждого  $i$  ( $1 \leq i \leq n$ ), станция  $i$  имеет уровень  $l_i$ . Конечные станции, а именно станция с номером 1 и  $n$  имеют уровень  $k$ .

Поезд типа  $j$  ( $1 \leq j \leq k$ ) останавливается на каждой станции, уровень которой больше или равен  $j$ , и он не останавливается на любых других станциях. Поскольку конечные станции, а именно станция 1 и станция  $n$ , имеют уровень  $k$ , каждый поезд останавливается на этих станциях.

Многие пассажиры ежедневно используют JOI Railways. Во время поездки они могут сесть на поезд, который идет в противоположном направлении от станции назначения, или они могут проехать нужную станцию. В конце поездки они должны достигнуть нужной станции. Они не очень любят останавливаться на станциях лишний раз. Следовательно, они пытаются пройти маршрут с минимальным количеством промежуточных станций. Станции, которые мы проезжаем мимо не являются промежуточными. Если пассажир останавливается на станции, чтобы поменять поезд, мы считаем это одной остановкой. Первая остановка на стартовой станции и последняя остановка в пункте назначения не считаются промежуточными станциями.

Ваша задача — написать программу, которая отвечает на запросы, какое минимальное количество промежуточных остановок получится для каждого пассажира.

### Формат входных данных

Первая строка содержит 3 целых числа  $n$ ,  $k$ ,  $q$  ( $2 \leq n \leq 100\,000$ ;  $1 \leq k \leq n$ ;  $1 \leq q \leq 100\,000$ ).  $n$  количество станций на JOI Railways, там есть  $k$  типов поездов, и есть  $q$  запросов про поездку между двумя станциями.

Далее дано  $n$  строк, где  $i$ -я строка ( $1 \leq i \leq n$ ) содержит число  $l_i$  ( $1 \leq l_i \leq k$ ), уровень станции  $i$ .

Затем дано  $q$  строк, где  $k$ -я строка ( $1 \leq k \leq q$ ) содержит два целых числа  $a_i$ ,  $b_i$  ( $1 \leq a_i, b_i \leq n$ ;  $a_i \neq b_i$ ), которые обозначают начальную и конечную станцию для пассажира с номером  $k$ .

### Формат выходных данных

Выведите  $q$  строк, где  $k$ -я строка ( $1 \leq k \leq q$ ) содержит минимальное количество промежуточных станций от станции  $a_k$  до станции  $b_k$ .

### Система оценки

Подзадача 1 (5 баллов):  $n, k \leq 100$ ;  $q \leq 50$

Подзадача 2 (15 баллов):  $q \leq 50$

Подзадача 3 (25 баллов):  $k \leq 20$

Подзадача 4 (55 баллов): Нет дополнительных ограничений.

## Примеры

стандартный ввод	стандартный вывод
9 3 3 3 1 1 1 2 2 2 3 3 2 4 4 9 6 7	1 3 0
5 2 1 2 1 1 1 2 1 4	1
15 5 15 5 4 1 2 3 1 1 2 4 5 4 1 5 3 5 8 1 11 1 5 3 6 11 9 12 15 14 15 2 3 12 2 1 4 8 15 5 12 6 1 13 13 8 14 9	2 1 1 3 2 0 3 4 0 1 3 4 1 2 2 2 1 1 2 2

## Замечание

В примере номер 1 приведены три запроса о маршрутах между станциями.

- Первый запрос со станции 2 до станции 4. Если пассажир выберет поезд типа 1 со станции 2 до станции 4, то будет одна промежуточная станция с номером 3, на которой остановится поезд.
- Второй запрос касается проезда от станции 4 до станции 9. Если пассажир выберет поезд с типом 1 со станции 4 до станции 5, потом сядет на поезд типа 2 со станции 5 до станции 1, потом на поезд типа 3 со станции 1 до станции 9, получится 3 промежуточных станции: 5, 1, 8.
- Третий запрос касается проезда от станции 6 до станции 7. Если пассажир выберет поезд типа 2 со станции 6 до станции 7, то не будет промежуточных станций.

Обратите внимание, что пассажиры могут проехать конечную станцию во время поездки.

## Задача В. Длинный замок

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

В одной компьютерной игре игрок попадает в длинный замок. В замке  $n$  комнат, расположенных в ряд и пронумерованных от 1 до  $n$ .

Для каждого  $i$  от 1 до  $n - 1$ , комнаты с номерами  $i$  и  $i + 1$  соединены двусторонним коридором. Для того, чтобы попасть из комнаты в коридор требуется ключ. Для каждого ключа определен тип. Может существовать более одного ключа каждого типа.

Для того, чтобы попасть из комнаты  $i$  или  $i + 1$  в соединяющий их коридор, необходим ключ типа  $c_i$ . В комнате с номером  $i$  лежат  $b_i$  ключей с типами  $a_{i,j}$  ( $1 \leq j \leq b_i$ ). Если игрок попадает в комнату, он подбирает все ключи в этой комнате. После этого он может использовать их, чтобы попасть в другие комнаты. Каждый ключ можно использовать более одного раза.

Для того, чтобы оптимизировать игровой процесс, требуется уметь отвечать на запрос следующего типа:

- Определить, может ли игрок попасть в комнату с номером  $y$  из комнаты с номером  $x$ , не имея изначально никаких ключей.

Для каждого из  $q$  запросов в вышеописанном формате, определите, сможет ли игрок из комнаты  $x$  попасть в комнату с номером  $y$ , не имея изначально никаких ключей.

### Формат входных данных

Первая строка входных данных содержит единственное целое число  $n$  — количество комнат в замке ( $2 \leq n \leq 500\,000$ ).

Вторая строка содержит  $n - 1$  целых чисел  $c_1, c_2, \dots, c_{n-1}$  — соответствие между коридором, соединяющим комнаты  $i$  и  $i + 1$ , и типом ключа  $c_i$ , необходимым для того, чтобы попасть в этот коридор ( $1 \leq c_i \leq n, 1 \leq i \leq n - 1$ ).

Каждая из следующих  $n$  строк содержит целое число  $b_i$ , а затем  $b_i$  целых чисел  $a_{i,1}, a_{i,2}, \dots, a_{i,b_i}$  — количество ключей в комнате  $i$  и сами типы ключей ( $1 \leq a_{i,j} \leq n; 1 \leq i \leq n, 1 \leq j \leq b_i \leq n; 1 \leq b_1 + b_2 + \dots + b_n \leq 500\,000$ ). Все  $b_i$  чисел  $a_{i,1}, \dots, a_{i,b_i}$  попарно различны ( $1 \leq i \leq n$ ).

Следующая строка содержит целое число  $q$  — количество запросов ( $1 \leq q \leq 500\,000$ ).

Каждая из следующих  $q$  строк содержит два целых числа  $x_k, y_k$  — комната, в которой находится игрок изначально и комната, в которую необходимо попасть в  $i$  запросе ( $1 \leq x_k \leq n, 1 \leq y_k \leq n; 1 \leq k \leq q; x_k \neq y_k$ ).

### Формат выходных данных

Выведите  $q$  строк. Для  $k$ -го запроса выведите в отдельной строке слово YES, если существует способ попасть из комнаты  $x_k$  в комнату  $y_k$  считая, что игрок сейчас в комнате  $x_k$  и не имеет никаких ключей.

### Система оценки

Подзадача 1 (5 баллов):  $n \leq 5\,000, q \leq 5\,000, b_1 + b_2 + \dots + b_n \leq 5\,000$ .

Подзадача 2 (5 баллов):  $n \leq 5\,000, b_1 + b_2 + \dots + b_n \leq 5\,000$ .

Подзадача 3 (15 баллов):  $n \leq 100\,000, c_i \leq 20$  ( $1 \leq i \leq n - 1$ ),  $a_{i,j} \leq 20$  ( $1 \leq i \leq n, 1 \leq j \leq b_i$ ).

Подзадача 4 (75 баллов): Нет дополнительных ограничений.

## Примеры

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

## Замечание

Пояснение к первому примеру:

- В первом запросе если игрок посетит комнаты в порядке 2, 1, 2, 3, 4, то доберется до комнаты 4.
- Во втором запроса игрок может попасть только в комнаты с номерами 3 и 4. Доступными являются только ключи 1 и 3 типов, игрок не может попасть в комнату 2.
- В третьем запросе игрок не может добыть ключ типа 4 для того, чтобы попасть из комнаты 4 в комнату 5. Следовательно, он не может попасть в комнату 5.
- В четвертом запросе, если игрок посетит комнаты в порядке 5, 4, 3, то попадет в комнату 3.

## Задача С. Междугородний автобус

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

Автобус курсирует между городом *from* и городом *to*. В автобусе есть резервуар с водой, из которого пассажиры и водитель могут пить во время пути. Автобус находится в городе *from* в момент времени 0 и добирается до города *to* за время  $x$ . На пути автобуса есть  $n$  заправочных станций, на которых можно наполнить резервуар водой. Возле  $i$ -ой заправки ( $1 \leq i \leq n$ ) автобус окажется через время  $s_i$ .

В начале пути в резервуаре нет воды, но его можно наполнить перед отправлением. Также резервуар можно наполнить на заправочной станции, стоимость воды в любой момент составляет  $w$  рублей за литр.

В городе *from*  $m$  пассажиров садятся в автобус. Пассажиры нумеруются от 1 до  $m$ . Нигде, кроме города *from*, пассажиры не садятся. Пассажиру  $j$  ( $1 \leq j \leq m$ ) потребуется литр воды во время  $d_j$ . После того, как пассажир выпьет воду, он захочет пить по прошествии времени  $t$ . Другими словами, пассажиру  $j$  требуется вода в моменты времени  $d_j + k \cdot t$  ( $k = 0, 1, 2, \dots$ ), где  $1 \leq d_j < t$  и значение  $t$  одинаково для всех пассажиров. Если в резервуаре недостаточно воды, когда пассажир хочет пить, пассажир сходит с автобуса. Если  $j$ -ый пассажир сходит с автобуса, не доезжая до города *to*, водитель платит штраф в размере  $c_j$  рублей. Сам водитель также нуждается в воде, и, как и другие пассажиры, после утоления жажды водитель снова захочет пить спустя время  $t$ . Другими словами, водителю нужна вода в моменты времени  $k \cdot t$  ( $k = 0, 1, 2, \dots$ ). Если в резервуаре автобуса нет воды, когда она требуется водителю, автобус прекращает свое движение.

Никакие два человека не нуждаются в воде в один и тот же момент времени. Также в моменты прибытия на заправку или в конечный пункт никто не нуждается в воде.

Регулируя количество воды в резервуаре, требуется минимизировать суммарную стоимость воды и штрафов, обеспечивающую завершение поездки в городе *to*. Ваша задача понять, где и каким количеством воды необходимо пополнять резервуар в течение путешествия, и найти минимальную стоимость воды и штрафов.

### Формат входных данных

Первая строка содержит 5 целых чисел  $x, n, m, w, t$  ( $1 \leq x \leq 10^{12}$ ,  $1 \leq n, m \leq 200\,000$ ,  $1 \leq t \leq x$ ,  $1 \leq w \leq 1\,000\,000$ ).

Далее идут  $n$  строк, где в  $i$ -ой ( $1 \leq i \leq n$ ) находится целое число  $s_i$  ( $1 \leq s_i < x$ ).

Далее в  $m$  строках задано два целых числа  $d_j$  и  $c_j$  ( $1 \leq d_j < t$ ;  $1 \leq c_j \leq 10^9$ ). Все  $d_j$  различны.

### Формат выходных данных

Выведите одно целое число — минимальную суммарную стоимость.

### Система оценки

Подзадача 1 (16 баллов):  $n, m \leq 8$

Подзадача 2 (30 баллов):  $n, m \leq 100$

Подзадача 3 (25 баллов):  $n, m \leq 2\,000$

Подзадача 4 (29 баллов): Нет дополнительных ограничений.

## Примеры

стандартный ввод	стандартный вывод
19 1 4 8 7 10 1 20 2 10 4 5 6 5	103
105 3 5 9 10 59 68 71 4 71 6 32 7 29 3 62 2 35	547
1000000000000 1 1 1000000 6 999999259244 1 123456789	333333209997456789

## Замечание

В первом примере: если залить в резервуар 7 литров воды перед отправлением и 4 литра на первой заправке, работа автобуса будет происходить следующим образом:

1. Когда автобус начнет движение в городе *from*, в резервуаре будет 7 литров воды.
2. Водитель и пассажиры 1, 2, 3, 4 выпьют по 1 литру воды в моменты времени 0, 1, 2, 4, 6, соответственно. После чего, в резервуаре останется 2 литра воды.
3. Водитель и пассажир 1 выпьют по 1 литру воды в моменты 7 и 8, соответственно. После чего, в автобусе не останется воды.
4. В момент 9 пассажир 2 захочет пить и будет вынужден покинуть автобус, так как в резервуаре в этот момент не будет воды.
5. В момент времени 10, мы зальем 4 литра воды на первой заправке. В резервуаре будет 4 литра воды.
6. Пассажиры 3, 4, водитель и пассажир 1 выпьют по 1 литру воды в моменты 11, 13, 14, 15, соответственно. В резервуаре не останется воды.
7. В момент 18 пассажир 3 захочет пить и покинет автобус из-за недостатка воды в резервуаре.
8. В момент 19 водитель прибудет в город *to*.

Количество купленной воды будет равно 11 литрам. Стоимость воды составит 88 рублей. Стоимость штрафов за пассажиров 2 и 3 составит 15 рублей. Суммарная стоимость будет равна 103 рублям.