

Георги се намира в хотел “Тракия” в Пазарджик и трябва да излезе, за да отиде до магазина. През това време Емил тъкмо влиза в хотела и тръгва нагоре по стълбите. Може да се каже, че Георги и Емил са в добри отношения, но не са първи приятели и понеже Георги е срамежлив и много бърза, той не иска да се засича с Емил по пътя към изхода.

В хотела има **N** локации, между които двамата герои могат да се придвижват - това може да са стаи, фойета на различните етажи и тем подобни. Между тях има **M** връзки - отрязъци от коридори и стълбища, като всички връзки са дълги точно по един метър. Георги не знае накъде се е запътил Емил, но знае че той изминава един метър за точно **a** секунди. Георги също така знае собствената си скорост - той изминава един метър за точно **b** секунди. Скоростта и на двамата не зависи от това дали се движат по коридора или нагоре/надолу по стълбите. Да, ясно ни е, че при нормални обстоятелства хората се движат с повече от един метър в секунда, но в тази задача действието се развива в късните часове на нощта и двамата ни герои са много уморени, затова се движат по-бавно. Както и да е, Георги се нуждае от такъв маршрут, че да не е възможно да се засече с Емил в някоя точка от него. Той може да излезе от хотела през един от **K** изхода, които се намират в локации $e_1, e_2, \dots, e_K$, а Емил влиза в хотела от първия изход, т.е. локация e_1 (“ха-ха, влиза от изход” - ами очевидно изходите са и входи). Помогнете на Георги да излезе от хотела!

Вход (escape.in)

В един тест ще трябва да решите няколко отделни сценария. От първия ред на входа се въвежда броят на сценариите **T**. За всеки сценарий данните се въвеждат по следния начин: на първия ред се въвеждат 5 естествени числа **N**, **M**, **a**, **b** и **s** - броя локации в хотела, броя връзки между тях, времената (в секунди) за които Емил и Георги съответно изминават по един метър и номерът на локацията, на която Георги се намира в момента. От следващите **M** реда се въвеждат по две числа - номерата на локациите, между които се намира съответната връзка. На следващия ред се въвежда числото **K** - броят изходи. Накрая се въвеждат на един ред **K** числа - номерата на локациите, в които се намират изходите.

Изход (escape.out)

За всеки сценарий изведете следното: ако Георги не може да излезе от хотела, така че да е сигурен, че няма да се засече с Емил, изведете на единствен ред числото **-1**. В противен случай, на първия ред на изхода изведете числото **L** - броя локации, през които минава маршрутът на Георги. На втория ред изведете **L** различни номера на локации, описващи маршрута. Първата трябва да съвпада с началното местоположение на Георги, а последната трябва да е изход. Между

всеки две локации, съседни в маршрута, трябва да има пряка връзка. Ако има повече от един възможен маршрут, изведете който и да е от тях.

Ограничения

$1 \leq N, M \leq 1\,000\,000$

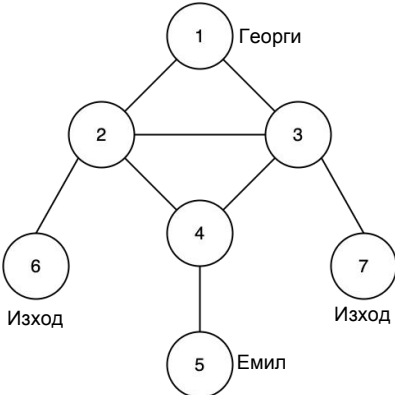
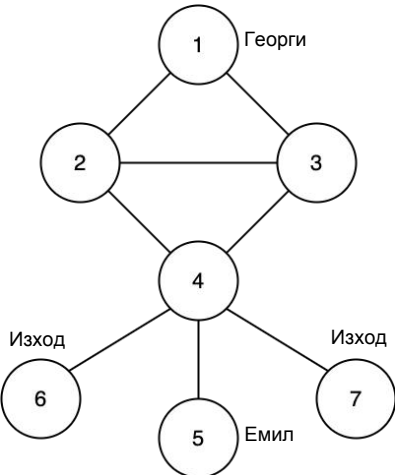
$1 \leq a, b \leq 10^9$

$1 \leq K, s, e_i, L \leq N$

$1 \leq T \leq 10$

Сборът на N, M във всички сценарии $\leq 1\,000\,000$

Пример

Вход	Изход	Обяснение
2 7 8 3 2 1 1 2 1 3 2 3 2 4 3 4 4 5 2 6 3 7 3 5 6 7 7 8 3 2 1 1 2 1 3 2 3 2 4 3 4 4 5 4 6 4 7 3 5 6 7	3 1 2 6 -1	Сценарий 1:  Сценарий 2: 

Connect 'Em

СЕЗОН 10 – ВТОРИ РУНД



Мишо обича да свързва квадратчета – това се е превърнало в любимото му занимание, когато скучае. Той разполага с разграфен лист хартия с N реда и M колони, номерирани съответно с числата от 1 до N и от 1 до M . В началото той поставя молива си в центъра на квадратчето с координати $(1, 1)$. След това може да го премести в центъра на $(1, 2)$ или $(2, 1)$. В общия случай, ако в момента върхът на молива се намира клетка с номер на реда i и номер на колоната j , Мишо може да го премести в $(i, j + 1)$ или $(i + 1, j)$. Целта му е, без да вдига молива си, да стигне до квадратчето в последния ред и последната колона т.е. това с координати (N, M) .

Веднъж докато му беше скучно в час, Мишо извади листчето и започна да свързва квадратчета. Неговата учителката забеляза това и се ядоса. За да не му пише забележка, тя му даде списък от K различни квадратчета и му постави условие да свърже така клетките, че да моливът да премине през всяко от тях. За съжаление, това невинаги е възможно. Напишете програма, която намира през колко от дадените квадратчета най-много може да премине моливът на Мишо, както и колко различни начина за такова изчертаване има. Два начина на изчертаване се считат за различни, ако единият преминава през дадено квадратче, а другият – не. Програмата трябва да обработва T тестови случая.

Вход

От първия ред на входния файл `connectem.in` се въвежда едно число T . Следват описания на T тестови случая.

От първия ред от описанието на всеки случай се въвеждат три числа – N , M и K . От следващите K реда се въвеждат по две числа R_i и C_i – координатите на клетките, които е дала учителката. Сред тях задължително присъстват $(1, 1)$ и (N, M) .

Изход

За всеки тестов случай на един ред от изходния файл `connectem.out` изведете две числа, разделени с един интервал - максималния брой квадратчета, които могат да бъдат свързани, и броя различни начини, по които може да бъде постигнато това.

Ограничения

$$1 < N, M \leq 100\,000$$

$$1 < K \leq 5\,000$$

$$1 \leq R_i \leq N$$

$$1 \leq C_i \leq M$$

Сумата от стойностите на K за всички тестови случаи няма да надхвърля 5 000.

Пример

Вход	Исход
1 4 5 5 1 1 1 2 2 4 4 3 4 5	4 13

Днес местният млекар Моньо ще напълни багажника на джипа си с мляко и ще тръгне да обикаля домовете на хората, да го продава (не джипа). В селата, през които Моньо минава (да, бизнесът му е доста обширен, не се задържа в едно единствено село) има общо N къщи, номерирани от 1 до N , и $N-1$ двупосочни улици, като пътувайки по улиците може да се стигне от всяка къща до всяка друга. Моньо живее в къща номер 1, а във всяка от останалите живее по един клиент. Има два типа клиенти: редовни и не толкова редовни. Всеки от редовните със сигурност ще купи едно шише мляко първия път, когато Моньо мине през къщата му. Нередовните клиенти купуват мляко от него само ако той им предложи, но в крайна сметка никой не му отказва, защото всеки е чувал колко е хубаво млякото му. Редовните клиенти са му по-ценни с това, че препоръчват на приятелите си от млякото и така помагат за разширяването на бизнеса. Той обаче се старее и сам да разраства бизнеса си, като привлича все повече и повече редовни клиенти. За целта Моньо иска да продава колкото се може повече на тези, които все още не са редовни, за да ги привлече, дори да се наложи да избягва редовните (все пак те го уважават наистина много и няма да му се разсърдят), само че не може да отказва на редовни клиенти, когато минава през тях.

Млекарят ще тръгне по домовете на хората по следния начин: ще започне от своята къща и на всяка минута ще отива в друга, която е свързана чрез улица с къщата, в която е в момента. Ако още не е минал през тази къща, той ще продаде една бутилка мляко на клиента, който живее в нея (ако клиентът е редовен, той със сигурност ще иска да си купи, в противен случай Моньо ще му продаде мляко, за да го привлече като редовен). Ако вече е минавал през тази къща, той ще я подмине. Когато остане без мляко, ще се прибере вкъщи. Помогнете му да определи на колко най-много нередовни клиенти може да продаде мляко, в зависимост от това колко шишета е натоварил.

Вход (monyo.in)

На първия ред се въвежда числото N . На следващите $N-1$ реда се въвеждат по две числа **A** и **B** - номерата на две къщи, между които има улица. На последния ред се въвежда низ от нули и единици с дължина $N-1$. Ако i -тия символ от низа е 1, в къща $i+1$ живее редовен клиент, а ако е 0 - нередовен.

Изход (monyo.out)

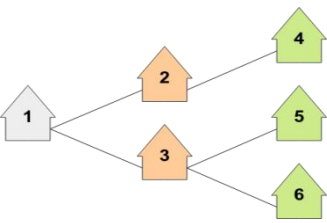
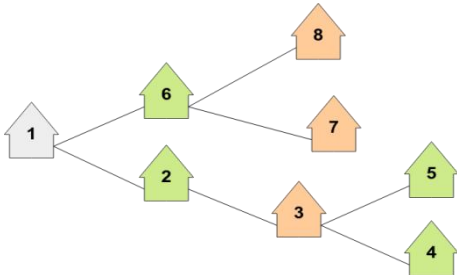
За всяко число i от 1 до $N - 1$ изведете на отделен ред на колко най-много нередовни клиенти Моньо може да продаде мляко, ако е заредил i шишета в багажника на джипа.

Ограничения

$1 \leq N \leq 1\,000\,000$

Броят редовни клиенти е до 1 000.

Примери

Вход	Изход	Обяснение
6 1 2 1 3 2 4 3 5 3 6 11000	0 1 2 2 3	 На картинката къщата на Моньо е оцветена в сиво, къщите на редовни клиенти в оранжево, а на нередовните - в зелено. Ако има само едно шише, Моньо ще го продаде или в къща 2, или в къща 3. Ако има две шишета, може например да продаде едното в къща 2 и другото в къща 4. Ако има 3 шишета, Моньо може да ги продаде в къщи 3, 5 и 6. Ако има 4 шишета, ще трябва мине и през къща 2, и през къща 3, затова ще може да продаде само 2 от шишетата на нередовни клиенти. Ако има 5 шишета, ще продаде по едно във всяка къща.
8 1 2 2 3 3 4 3 5 1 6 6 7 6 8 0100011	1 2 2 3 4 4 4	 Ако има 1 или 2 шишета, Моньо може да ги продаде в къщи 2 и 6, без да минава през нито един редовен клиент. Ако има 3, 4 или 5 шишета, може да ги продаде като минава през само един редовен клиент (този в къща 3). Ако има 6 или 7 ще продаде последните в къщи 7 и 8.

Network visibility



СЕЗОН 10 – ВТОРИ РУНД

Светльо реши да се занимава с компютърни мрежи. Компютърна мрежа наричаме съвкупност от N устройства (номерирани с числата от 1 до N), свързани с $N - 1$ кабела, която има йерархична структура – всяко устройство с изключение едно (наречено главно устройство) има пряк родител, от който получава ip адрес.

По време на заниманията си той се сблъска със задачата да се определи колко устройства в мрежата са видими от дадено устройство. Казваме, че едно устройство v е видимо от друго устройство u , тогава и само тогава когато е изпълнено едно от следните условия:

- 1) u е родител на v в йерархичната мрежа
- 2) v е предшественик (непряк родител) на u в йерархичната мрежа
- 3) родителят на v е предшественик на u в йерархичната мрежа

Наскоро Светльо откри една нова функционалност на мрежовите устройства – режима "access point" (AP). Когато едно устройство е в този режим, то действа само като физическа връзка между другите устройства и на практика е невидимо в мрежата. Освен това, устройствата, на които е било родител вече се свързват директно към най-близкия си предшественик, който не е в AP, за да получат ip адрес. Разбира се, главното устройство не може да бъде в режим AP.

За да продължи да изучава свойствата на мрежите на Светльо му е необходима програма, която обработва Q заявки от следните видове:

- 1) **$T\ x$** – променя режима на устройство с номер x (ако AP е включен – го изключва, и обратно)
- 2) **$V\ x$** – устройството с номер x се поврежда (или ако вече е повредено бива заменено с ново). При повреда на устройство, мрежата се разпада и се създават нови подмрежи, всяка от които има за главно едно устройство, чийто родител е бил x . Ако някое от тези главни устройства е в режим AP, той се изключва. Също така, когато повредено устройство се заменя с ново, децата му се свързват отново към него. То и неговите деца не са в режим AP (дори и да са били преди повредата).
- 3) **$S\ x$** – преброява видимите устройства от устройство с номер x в мрежата. Въпросното устройство със сигурност не е в режим AP.

Вход

На първия ред на входния файл `visibility.in` са зададени числата N и Q . На втория ред са зададени N числа, i -тото от които е номерът на родителя на устройството номер i или 0, ако то е главно устройство. Всеки от следващите Q реда описва по една заявка в гореописания формат.

Изход

За всяка заявка от трети тип на отделен ред от изходния файл `visibility.out` изведете по едно число – търсения брой видими устройства.

Ограничения

$1 < N, Q \leq 100\,000$

$1 \leq x \leq N$

Пример

Вход	Изход
12 10	8
0 1 2 3 3 1 6 6 8 8 10 6	2
C 10	3
C 1	5
C 2	6
T 6	6
C 2	5
C 9	
T 3	
C 2	
B 1	
C 8	

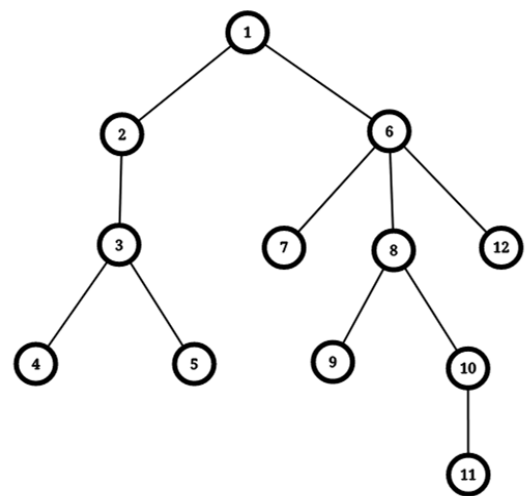
Пояснение

Началното състояние на мрежата е отразено на картинката вдясно.

След четвъртата заявка, устройство 6 става невидимо, а 7, 8 и 12 се свързват директно към 1. Така последните три стават видими за устройство 2.

След седмата заявка, устройство 3 става невидимо, а 4 и 5 се свързват директно към 2. По този начин от устройство 2 са видими 6 устройства – 1, 4, 5, 7, 8 и 12.

След деветата заявка, устройство 1 се поврежда и се създават две нови подмрежи, чиито главни устройства са съответно 2 и 6, като 6 преминава отново в нормален режим. В тази ситуация устройствата, видими от 8, са 6, 7, 9, 10 и 12.



Distillation

СЕЗОН 10 - ВТОРИ РУНД



Рая е ценителка на хубавото уиски. Тя е проучила N малки дестилерии, номерирани от 1 до N , и иска да инвестира в една от тях. Процесът на правене на уиски в тях протича на два етапа:

1. Първо се изпичат ечемикът и малцът, от тях се получава разтвор, който ферментира и впоследствие се дестилира, за да се получи уиски.

2. След това полученото уиски се оставя да отлежи - колкото по-отлежало е, толкова е по-хубаво.

Вторият етап не може да започне, преди да е приключил напълно първият, т.е. получава се определено количество уиски, приключва първият етап и чак тогава цялото уиски наведнъж се оставя да отлежи.

Рая е много стриктна към качеството на уискито - толкова е стриктна, че го измерва с числа. За дестилерия номер i тя си е записала 4 реални числа A_i , B_i , C_i и D_i , които означават следното:

1. Дестилерията се нуждае от A_i дни за подготовка, преди да започне да произвежда уиски.

2. През първия етап на производство в нея ще се получават B_i литра уиски с качество C_i всеки ден.

3. Като приключи първият етап и уискито започне да отлежава, неговото качество ще се подобрява с D_i единици на ден. Преди да се остави да отлежава, качеството на уискито не се променя.

За да избере в коя дестилерия да инвестира, Рая има Q въпроса към вас от следния вид: Ако всички дестилерии разполагат с T дни и си разпределят времето оптимално, колко е най-голямото възможно производство на количество (в литри) и качество, което някоя от тях може да получи, и коя дестилерия ще го получи? Забележете, че T не е задължително цяло число и процесите протичат равномерно, т.е. ако за един ден се произвеждат B литра уиски, за половин ден например ще се произведат $\frac{B}{2}$ и ако за един ден качеството на уискито нараства с D единици, то за половин ден ще нарастне с $\frac{D}{2}$.

Вход (distillation.in)

На първия ред се въвежда числото N - броят на дестилериите. На всеки от следващите N реда се въвеждат A_i , B_i , C_i и D_i - числата, които Рая си е записала за i -тата дестилерия. На следващия ред се въвежда числото Q - броят на заявките. На всеки от следващите Q реда се въвежда числото T_j - дните за j -тата заявка.

Изход (distillation.out)

За всяка заявка изведете на отделен ред две числа, разделени с интервал: най-голямото възможно произведение на литри и качество, което някоя дестилерия може да получи, и индекса на дестилерията, която ще го получи. Ако има няколко дестилерии, получаващи равно на най-голямото произведение, изведете най-малкия индекс на една от тях. Първото число ще се счита за правилно, ако абсолютната разлика между него и числото, получено от авторското решение за съответната заявка, е не по-голяма от 10^{-5} .

Ограничения

$$1 \leq N, Q \leq 100\,000$$

$$0 \leq A_i, C_i \leq 1\,000$$

$$0 < B_i, D_i, T_j \leq 1\,000$$

A_i, B_i, C_i, D_i и T_j се въвеждат с до 5 цифри след десетичната запетая.

Пример

Вход

```
2
3 2.5 1.5 2
1 1.25 0 1.25
3
2
3.5
6
```

Изход

```
0.390625 2
2.441406 2
17.578125 1
```