

Петокласничката Сашка я наричат “Криндж”, защото виси по цял ден в ТикТок. От твърде много видеа с Freddy Fazbear и Ванеса ѝ втръсна, заради това тя реши да се освежи със следната задача по информатика:

Даден ви е низ от нули и единици $s_1, s_2, s_3, \dots, s_N$, съставен от N символа. Вие може да избирате подниз от съседни елементи $s_l, s_{l+1}, s_{l+2}, \dots, s_r$ ($1 \leq l \leq r \leq N$) и да „преобърнете“ стойностите на елементите в него. По-точно, за всяко $l \leq i \leq r$, $s_i := 1 - s_i$. Вие трябва да намерите минималния брой операции, нужен за да се сортира низа възходящо. Възходящо сортиран низ е такъв, за който за всяко $2 \leq i \leq N$, $s_{i-1} \leq s_i$.

След много мъки тя успя да реши задачата по елегантен начин (даже брат ѝ беше впечатлен!). Сега е Ваш ред да напишете програма `freddy.cpp`, която решава споменатата задача.

Вход

На първия ред от `freddy.in` е дадено числото N . На втория ред от стандартния вход са дадени N символа, i -тия от тях е s_i .

Изход

На първия ред от `freddy.out` изведете едно число – минималния брой операции, нужен за възходящо сортиране на низа. Ако не е възможно, изведете -1 .

Ограничения

$$1 \leq N \leq 10^5$$

$$0 \leq s_i \leq 1$$

Ограничение по време: 0.2 сек.

Ограничение по памет: 256 МВ.

Примерни тестове

Вход (<code>freddy.in</code>)	Изход (<code>freddy.out</code>)
5 11111	0
5 10011	1
10 0101101011	3
20 01010001100010011010	6

Matrices



СЕЗОН 2021/2022 – ЧЕТВЪРТИ РУНД

Хари има много голяма матрица. Ако трябва да сме точни, матрица $N \times M$. Наскоро беше рожденият ден на Хари, като (за негова изненада) получи подарък от приятелите си – комплект с неотрицателни числа. Сега той по цял ден си играе да нарежда числата в матрицата. За да му бъде по-интересно, измислил следното правило: в клетка на i -ти ред j -та колона (редовете и колоните са номерирани от 1) не може да стои число по-голямо от $\min(i, j)$.

Понеже Хари много цени приятелите си, иска да ги зарадва, като им покаже **хубава матрица**. За него **хубава матрица** е матрица, в която сборът от числата във всеки ред и всяка колона е четно число (да, Хари е странен). На фигурата е даден пример за хубава матрица с размер 3×4 :

0	1	0	1
0	1	1	2
0	2	3	1

Хари обаче е максималист и една матрица не му стига. Той иска да знае колко са всички **хубави** матрици, които може да направи. За нещастие няма време да прави тези абсурдни сметки, затова моли вас да му помогнете, като намерите колко са всички матрици, удовлетворяващи условието.

Тъй като това число може да е изключително голямо, трябва да го изведете по модул от $1,000,000,007 (10^9 + 7)$.

Вход

От първия ред на файла `matrices.in` се въвежда едно цяло положително число T , задаващо броя на тестовите примери. На следващите T реда се задават размерите на матрицата N и M за съответния тест.

Изход

Във файла `matrices.out` трябва да отпечатате броя на **хубавите матрици** за всеки тест. Отговорите за отделните тестове трябва да са на отделни редове.

Ограничения

$$1 \leq N, M \leq 10^5$$

$$1 \leq T \leq 10$$

Ограничение по време: 1сec.

Ограничение по памет: 256 MB.

Matrices

СЕЗОН 2021/2022 – ЧЕТВЪРТИ РУНД



Примерен тест

Вход (matrices.in)	Изход (matrices.out)
4 2 3 4 2 1 5 134 231	9 27 1 716490517

Accounting

СЕЗОН 2021/2022 – ЧЕТВЪРТИ РУНД



След твърде много напрежение, Сашка реши да стане счетоводител. Нейната работа е да обработва двутипни заявки върху единствения масив на компанията $a_1, a_2, a_3, \dots, a_N$ от N елемента. Заявките са следните:

- $l \ r \ x$ – За всеки елемент a_i ($l \leq i \leq r$) от масива, $a_i := x - a_i$

? $l \ r$ – Въпрос за стойността на $a_l + a_{l+1} + a_{l+2} + \dots + a_r$.

Сашка си мрази работата, заради това тя реши с всичките си знания да направи програма, която да обработва заявките вместо нея. За жалост, програмата ѝ се оказала твърде бавна. Вие, като неин колега, сте загрижен за нея, поради което ще се опитате да напишете програма `accounting.cpp`, която да помогне на Сашка.

Вход

На първия ред от файла `accounting.in` се въвеждат две положителни числа N и Q , съответно броят числа в масива и броят на заявките. На следващия ред са дадени N числа, съответно $a_1, a_2, a_3, \dots, a_N$. На следващите Q реда са ви дадени заявките в гореописания формат.

Изход

За всяка заявка за въпрос изкарайте едно число на един ред в `accounting.out`, равно на отговора на заявката.

Ограничения

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

$$0 \leq a_i, x \leq 10^6$$

$$1 \leq l \leq r \leq N$$

Ограничение по време: 1.5 sec.

Ограничение по памет: 256 MB.

Accounting

СЕЗОН 2021/2022 – ЧЕТВЪРТИ РУНД



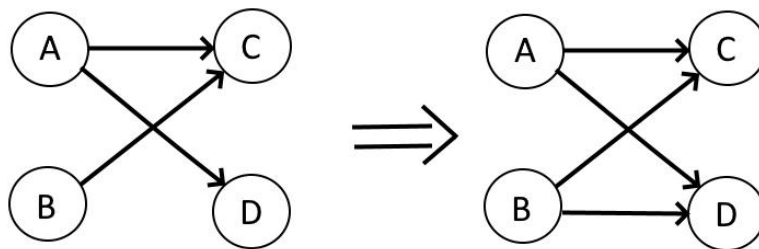
Примерни тестове

Вход (accounting.in)	Изход (accounting.out)
5 7 57 18 40 89 7 - 2 3 94 ? 1 5 ? 3 5 - 2 4 41 ? 2 4 ? 1 5 ? 3 4	283 150 -96 -32 -61
10 10 92 7 84 86 14 3 58 56 19 80 - 2 10 63 ? 3 10 ? 5 8 - 1 6 72 ? 3 6 - 2 8 32 ? 2 3 ? 9 10 ? 3 9 ? 4 8	104 121 223 -45 27 1 18

Кюшо решил, че е крайно време да си вземе почивка от целия стрес покрай управлението на фирмата и да поеме на семейно пътешествие. След задълбочено мислене набелязал **N** дестинации и **M** директни еднопосочни пътища, чрез които да се придвижва между тях. За да е удовлетворението от обиколката максимално, той искал да посетят всяка от дестинациите точно по веднъж и от последната да могат директно да се приберат вкъщи. Така съставил **T** разписания с дестинации и пътни отсечки между тях и тъй като времето е ограничено, моли Вас да определите за кои от тях съществува маршрут от описания вид. Една последна подробност, която Кюшо смята, че може да Ви е необходима, е странното свойство на всяко разписание:

Ако за дадени две дестинации **A** и **B** съществува дестинация **C**, такава че има директни пътища и от **A**, и от **B** към нея, то всяка друга дестинация **D**, до която има пътна отсечка от **A** или **B**, е като **C** (има директен път и от **A**, и от **B** към нея).

Или по друг начин казано, ако наричаме дестинация **Y** съсед на **X**, тогава и само тогава, когато от **X** има пътна отсечка към **Y**, то за всеки две дестинации е в сила, че ако имат поне един общ съсед, всичките им съседи са общи.



Вход

От първия ред на файла `eksk.in` се въвежда естественото число **T**. За всяко от разписанията се въвеждат **N** и **M**, следвани от **M** реда с по две числа **X** и **Y**, задаващи, че има еднопосочен път от дестинация **X** към дестинация **Y** ($0 \leq X, Y \leq N$; с **0** се отбелязва скромното име на Кюшо (за него също важи гореописаното свойство), а с числата от **1** до **N** – номерата на дестинациите).

Изход

За всяко от разписанията изведете във файла `eksk.out` на отделен ред **YES**, ако има маршрут от търсения вид или **NO** в противен случай. Ако съществува, на следващия ред изведете и **N+2** числа (първото и последното от които **0**), описващи дестинациите по реда им на посещаване. Ако има няколко възможни маршрута, изведете кой да е от тях.

Ограничения

$$1 \leq T \leq 10^2$$

$$1 \leq \text{сумата на всички } N \leq 10^5$$

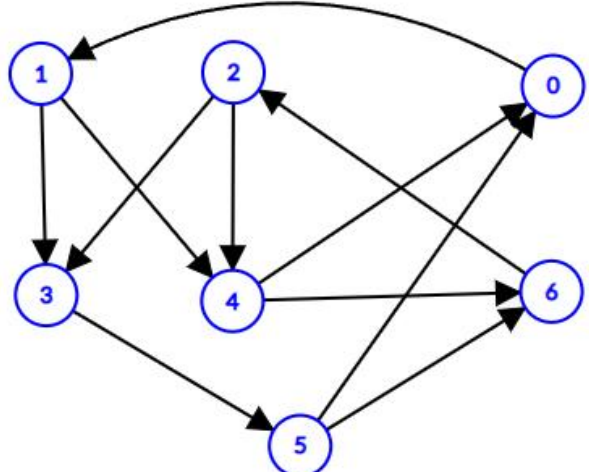
$$1 \leq \text{сумата на всички } M \leq 5 \times 10^6$$

$$0 \leq X, Y \leq N \text{ и } X \neq Y \text{ за всяка пътна отсечка}$$

Ограничение по време: 1.2 sec.

Ограничение по памет: 256 MB.

Примерен тест

Вход (eksk.in)	Изход (eksk.out)	Пояснение
<pre> 1 6 11 1 3 2 3 1 4 2 4 3 5 5 6 4 6 4 0 5 0 0 1 6 2 </pre>	<pre> YES 0 1 4 6 2 3 5 0 </pre>	

Сашка обича да ходи редовно всеки четвъртък на школата Ви по програмиране, където често изучавате заедно интересни теми. Миналия път Сашка изучава със заместник побитови операции, като обърнаха особено внимание на операцията *изключващо или* (XOR). Тя, впечатлена от способностите на тази операция, реши да отвори тефтера на по-големия си брат, в който има записани много интересни задачи. Първоначално успяла да реши по-лесните, но се спряла на една много екзотична задача. Условието гласяло следното:

Дадено Ви е дърво от N върха, с тегла по ребрата. Позволено Ви е да направите K промени по стойностите на ребрата, така че да минимизирате броя на простите *неприятни* пътища в дървото. Прост път в дърво е път, който не включва връх повече от веднъж. *Неприятен* път е такъв, за който *изключващото или* на стойностите на ребрата, участващи в него, има нечетен брой единици в побитовото му представяне. Намерете този минимален брой *неприятни* пътища след оптимално прилагане на операциите.

Сашка много дълго време мислила и творила по задачата, и в крайна сметка успяла да я реши! Поне така мислила, докато не дошъл следващият четвъртък, когато Вие сте ѝ казали, че тя винаги прави $K + 1$ операции. Сега тя е много нещастна, заради това се захващате да решите задачата, за да успокоите Сашка.

Вход

На първия ред от файла `xorfun.in` се въвеждат две положителни числа N и K . На следващите $N - 1$ реда се въвеждат по 3 естествени числа – двата края на ребро от дървото v_i и u_i , както и неговото тегло w_i .

Изход

На един ред от `xorfun.out` изведете едно число – минималния възможен брой *неприятни* пътища.

Ограничения

$$1 \leq N \leq 5000$$

$$1 \leq K \leq 500$$

$$1 \leq v_i, u_i \leq N$$

$$0 \leq w_i \leq 2^{31} - 1$$

Ограничение по време: 3 сек.

Ограничение по памет: 256 MB.

Примерни тестове

Вход (xorfun.in)	Изход (xorfun.out)
9 3 1 2 6 1 7 4 2 3 7 3 4 12 3 5 14 3 6 13 5 8 2 5 9 11	8
10 3 5 2 938707311 6 5 312182765 7 5 952433887 1 5 1630822531 3 5 1850473008 9 1 510193547 8 6 1465047925 10 8 1237069467 4 9 1320045850	9
15 4 1 3 1936267205 3 10 1050608599 15 1 1084713226 15 6 327822164 10 8 1085192654 1 9 479650095 12 9 16193632 12 11 1114880438 8 5 1011320449 8 13 2012210084 14 3 47220444 7 15 1406284003 4 1 16038788 13 2 733939625	36