

CEOI 2026



Ден 2
(български)

Day 2
(Bulgarian)

Рязане на цветя

Ограничение по време: 4 s

Ограничение по памет: 256 MiB

В общата СЕОІ градина отглеждаме колекция от специални цветя, които плътно преплитат корените си. Когато ги режем, те израстват отново, стига да не режем твърде агресивно, за да не причиняваме непоправими щети.

Ако двойка цветя a и b са преплели корените си, казваме, че те са „свързани“. В противен случай те са „несвързани“.

Корените се разрастват според следното правило: Нека a и b са две несвързани цветя. Ако съществуват поне още 2 цветя c и d , така че и a , и b да са свързани както с c , така и с d , тогава между a и b ще израснат корени и и те ще станат свързани.

Тези цветя растат вече от известно време и всички корени, които биха могли да израснат според горното правило, вече са израснали. С други думи, ако цветя a и b са (и двете) свързани с някои две цветя c и d , тогава е гарантирано, че a и b също са свързани помежду си.

Трябва да изкореним тази градина и да я преместим при мястото на следващото СЕОІ. За да улесним преместването, бихме искали да срежем възможно най-много корени. Въпреки това искаме цветята да успеят да израснат отново до сегашното си състояние. Какъв е максималният брой свързани двойки цветя, чиито връзки можем да прекъснем, така че те все пак да израснат отново до сегашното си състояние? Няма значение колко итерации на растеж ще са необходими.

Вход

Първият ред на входа съдържа две цели числа n и m , разделени с интервал — броят на цветята и броят на съществуващите връзки между тях. Следват m реда, всеки съдържащ двойка цели числа a_i и b_i , указващи, че цветята a_i и b_i са свързани.

Цветята са номерирани с цели числа от 1 до n .

Гарантирано е, че входните данни удовлетворяват правилото, описано в условието на задачата.

Ограничения

- $1 \leq n \leq 1000$
- $1 \leq m \leq 10^5$

Подзадачи

- Подзадача 1 (20 точки): $n \leq 10$ и $m \leq 20$
- Подзадача 2 (14 точки): $m = \frac{n \cdot (n-1)}{2}$
- Подзадача 3 (15 точки): Гарантира се, че всяко цвете е свързано с най-много 7 други цветя.
- Подзадача 4 (15 точки): $n \leq 50$ и $m \leq 1000$
- Подзадача 5 (36 точки): Няма допълнителни ограничения.

Изход

Изведете едно цяло число — максималния брой връзки, които можем да прекъснем.

Пример

Вход

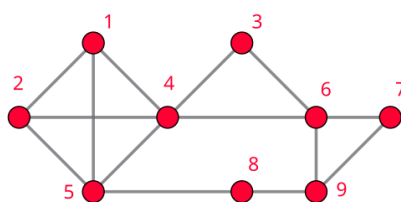
9 14
1 2
1 4
1 5
2 4
2 5
3 4
4 5
3 6
4 6
6 7
6 9
7 9
8 9
5 8

Изход

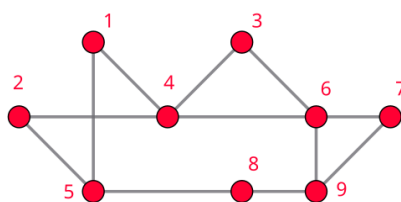
2

Коментар

Цветята в примера преди каквото и да е рязане. Лесно може да се провери, че не могат да се образуват допълнителни връзки според описаното правило за растеж.



Цветята в примера след рязането имат с 2 връзки по-малко. Връзката между 1 и 2 може да израсне отново, защото и двете цветя 1 и 2 са свързани с цветята 4 и 5. По същия начин връзката между 4 и 5 може да израсне отново, защото цветята 4 и 5 са свързани с цветята 1 и 2.



Кули

Ограничение по време: 1 s Ограничение по памет: 256 MiB

Имате n компютъра и m кули, всички разположени на различни позиции на една права линия. Трябва да групирате компютрите по двойки с кабели по такъв начин, че всеки кабел да започва от единия компютър, да посещава някои кули и да завършва в другия компютър. Кабелът може да посети които и да е от кулите (не само тези между компютрите), в какъвто и да е ред. Той може да пропуска кули, като преминава покрай тях, без да ги посещава. Също така може да не посещава нито една кула и да свърже двата компютъра директно, но не може да свързва компютър със самия себе си. Броят на компютрите е четен.

Нека a и b са позициите на два групирани компютъра, а $x_1, \dots, x_k$ са позициите на кулите, които кабелът посещава. Дължината на кабела е $|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|$. За даден кабел дефинираме неговата оценка като $f \cdot u - l$, където l е неговата дължина, f е фиксирана константа, а u е броят на уникалните кули, които кабелът посещава измежду $x_1, \dots, x_k$. Няколко кабела могат да посещават една и съща кула и тази кула допринася към оценката на всеки от тези кабели.

Трябва да изчислите максималната възможна сума от оценки на кабелите при групиране на компютрите по двойки (т.е. всеки компютър трябва да принадлежи точно на една двойка, или еквивалентно — да бъде свързан с точно един кабел).

Вход

Първият ред съдържа броя на тестовите случаи T . Следват тестовите случаи един след друг. Всеки тестов случай се състои от три реда. Първият ред съдържа три цели числа n , m и f — броя на компютрите, броя на кулите и константата f . Вторият ред съдържа n цели числа $a_1, a_2, \dots, a_n$ — позициите на компютрите. Третият ред съдържа m цели числа $b_1, b_2, \dots, b_m$ — позициите на кулите.

Ограничения

Нека N и M са съответно сумите на n и m във всички тестови случаи.

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $0 \leq f \leq 10^9$
- n е четно.
- $1 \leq a_i, b_i \leq 10^9$
- Всички позиции на компютрите и кулите са различни (в рамките на един тестов случай).

Подзадачи

- Подзадача 1 (5 точки): $N \leq 5000$, $m = 1$

- Подзадача 2 (10 точки): $T \leq 20$, $n \leq 10$, $m \leq 100$
- Подзадача 3 (27 точки): $N, M \leq 5000$
- Подзадача 4 (21 точки): $N \leq 5000$
- Подзадача 5 (37 точки): Няма допълнителни ограничения.

Изход

Изведете T цели числа, всяко на отделен ред — максималната възможна сума от оценките на кабелите за всеки тестов случай.

Пример

Вход	Изход
4	89
2 1 100	-4
1 10	12
11	51
4 1 10	
2 4 6 8	
20	
4 1 10	
2 4 6 8	
5	
6 3 10	
2 13 4 8 6 10	
5 1 9	

Vim

Ограничение по време: 1 s Ограничение по памет: 256 MiB

Ще е трудно да намерите нещо *по-демоде* от старомодния Vim — редактор, в който може да правим неща, за които само може да си мечтаем в "стандартен" редактор. Достатъчно е само да инвестираме една седмица да го изучим и един месец да го потренираме, така че неговите необичайни, но мощни, команди да станат част от "мускулната ни памет". Все пак, както във всички други редактори и във Vim имаме *курсор*, който винаги се намира върху някой от символите. В началото той се намира върху първия символ от текста. Подобно на другите редактори Vim има и *клипборд*, в който е предвидено да се запазва (и индиректно да се копират и преместват) части от текста. Клипбордът в началото е празен.

В тази задача ще приемем, че в началото в редактора има записан точно един символ '-' (минус). Нашата цел е накрая да има записани точно n последователни знака минус. Ще използваме следните четири команди за целта:

- **h**: Ако курсорът е върху първия символ, то тази команда няма да направи нищо. В противен случай, курсорът се премества върху символа вляво.
- **l**: Ако курсорът е върху последния символ, то тази команда няма да направи нищо. В противен случай, курсорът се премества върху символа вдясно.
- **Y**: Поредицата от символи, започващи от позицията на курсора до края на текста се копира в клипборда, "презаписвайки" предишното съдържание на клипборда. (Ако курсорът е върху първия символ на текста, то целият текст ще се запише в клипборда.)
- **P**: Копие на текста, записан в клипборда, се поставя преди символа, върху който е курсорът, а курсорът се премества на последния поставен символ. Съдържанието на клипборда не се променя. Ако клипбордът е празен, нищо не се променя.

Напишете програма, която прочита числото n и отпечатва минималния брой команди, нужни, за да има накрая записани точно n последователни '-' символа във Vim при поставените условия. Програмата трябва да отпечатва и примерна редица с минимален брой команди.

Вход

Първият ред съдържа броя на тестовите случаи t . Следващите t реда съдържат тестови случаи с търсената дължина n .

Ограничения

- $1 \leq t \leq 100$
- $1 \leq n \leq 10^7$

Подзадачи

Ако отпечатате само верния брой команди, но нямате пример или имате неправилна редица от команди, то ще получите половината от точките за подзадачата.

- Подзадача 1 (20 точки): $n \leq 100$
- Подзадача 2 (8 точки): $n \leq 1000$
- Подзадача 3 (18 точки): $n \leq 10^4$
- Подзадача 4 (18 точки): $n \leq 10^5$
- Подзадача 5 (18 точки): $n \leq 10^6$
- Подзадача 6 (18 точки): Няма допълнителни ограничения.

Изход

За всеки тестов случай отпечатайте на един ред минималния брой команди и пример за редица от команди на същия ред.

Пример

Вход

2
21
2

Изход

10 YURhYURPP
2 YP

Коментар

Както може да видим в следната таблица, в първия тестов случай редицата YURhYURPP дава правилен резултат. Символът '=' в колоната Екран задава символът '-', върху който се намира курсорът.

Стъпка	Команда	Екран	Клипборд
0		=	(празен)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	=---	--
5	h	=----	--
6	P	=-----	--
7	Y	=-----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----