

CEOI 2026



День 2
(Українська)

Day 2
(Ukrainian)

Обрізання мальв

Ліміт часу: 4 s Ліміт пам'яті: 256 MiB

На маминому ганку ростуть особливі мальви, які тісно переплітають свої корені. Якщо їх обрізати, вони відростуть знову, доки ми не обріжемо їх надто агресивно та не завдамо непоправної шкоди.

Якщо пара мальв a і b переплітає свої корені, ми кажемо, що вони «з'єднані». В іншому випадку вони «нез'єднані». Корені ростуть за таким правилом: Нехай a і b — дві нез'єднані мальви. Якщо існують принаймні 2 інші мальви c і d такі, що і a , і b з'єднані як із c , так і з d , тоді між a і b виростуть корені, і вони стануть з'єднаними.

Ці мальви вже ростуть протягом певного часу, і всі корені, які могли вирости відповідно до наведеного вище правила, вже вирости. Іншими словами, якщо дві мальви a і b обидві з'єднані з деякими мальвами c і d , то гарантовано, що a і b також з'єднані між собою.

Але, на жаль, нам необхідно пересадити мальви на інший ганок. Щоб спростити пересадку, потрібно перерізати якомога більше коренів. Проте ми хочемо, щоб після цього мальви знову відрости до свого поточного стану. Яку максимальну кількість з'єднань між парами мальв можна перерізати так, щоб вони все одно знову відрости до свого поточного стану? Не має значення, скільки ітерацій росту для цього знадобиться.

Вхідні дані

Перший рядок вхідних даних містить два розділені пробілом цілі числа n і m , — це кількість мальв і кількість наявних з'єднань між ними. Далі йдуть m рядків, кожен із яких містить пару цілих чисел a_i і b_i , що означає, що мальви a_i і b_i з'єднані. Мальви позначені цілими числами $1 \dots n$. Вхідні дані гарантовано відповідають правилу, описаному в умові задачі.

Обмеження

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

Підзадачі

- Підзадача 1 (20 балів): $n \leq 10$ та $m \leq 20$
- Підзадача 2 (14 балів): $m = \frac{n \cdot (n-1)}{2}$
- Підзадача 3 (15 балів): Кожна мальва гарантовано з'єднана не більш ніж із 7 іншими мальвами.
- Підзадача 4 (15 балів): $n \leq 50$ та $m \leq 1000$
- Підзадача 5 (36 балів): Без додаткових обмежень.

Вихідні дані

Виведіть одне ціле число — максимальну кількість з'єднань, які можна обрізати.

Приклад

Вхід

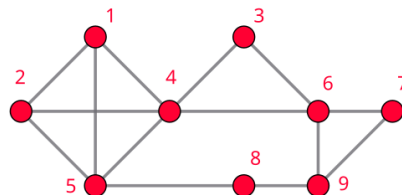
9 14
 1 2
 1 4
 1 5
 2 4
 2 5
 3 4
 4 5
 3 6
 4 6
 6 7
 6 9
 7 9
 8 9
 5 8

Вихід

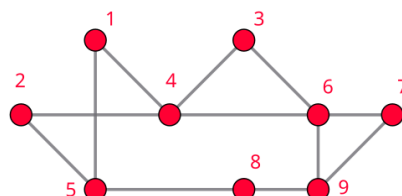
2

Примітка

Мальви у прикладі до будь-якого обрізання. Можна перевірити, що жодні додаткові з'єднання не можуть утворитися відповідно до описаної процедури росту.



Мальви у прикладі після обрізання мають на 2 з'єднання менше. З'єднання між 1 і 2 може відрости знову, тому що мальви 1 і 2 обидві з'єднані з мальвами 4 і 5. Аналогічно, з'єднання між 4 і 5 може відрости знову, тому що мальви 4 і 5 обидві з'єднані з мальвами 1 і 2.



Вежі

Ліміт часу: 1 s Ліміт пам'яті: 256 MiB

У вас є n комп'ютерів і m веж, розташованих у різних точках на прямій. Потрібно утворити пари комп'ютерів кабелями так, щоб кожен кабель починався в деякому комп'ютері, відвідував кілька веж і закінчувався в іншому комп'ютері. Кабель може відвідувати будь-які вежі (не лише ті, що розташовані між комп'ютерами) у довільному порядку. Він може пропускати вежі, проходячи повз них без відвідування. Також він може не відвідати жодної вежі й напряду з'єднати два комп'ютери, але не може з'єднувати комп'ютер із самим собою. Кількість комп'ютерів є парною.

Нехай a і b — позиції двох комп'ютерів, а $x_1, \dots, x_k$ — позиції веж, які відвідує кабель. Довжина кабелю дорівнює $|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|$. Для деякого кабелю визначимо його оцінку як $f \cdot u - l$, де l — його довжина, f — деяка фіксована константа, а u — кількість унікальних веж, які цей кабель відвідує серед $x_1, \dots, x_k$. Кілька кабелів можуть відвідувати ту саму вежу, і ця вежа враховується в оцінці кожного з цих кабелів.

Потрібно обчислити максимально можливу суму оцінок кабелів, утворивши пари з усіх комп'ютерів (тобто кожен комп'ютер має належати рівно одній парі або, еквівалентно, бути підключеним рівно одним кабелем)

Вхідні дані

Перший рядок містить кількість тестових випадків T . Далі йдуть тестові випадки один за одним. Кожен тестовий випадок складається з трьох рядків. Перший рядок містить три цілі числа n , m і f — кількість комп'ютерів, кількість веж і константу f . Другий рядок містить n цілих чисел $a_1, a_2, \dots, a_n$ — позиції комп'ютерів. Третій рядок містить m цілих чисел $b_1, b_2, \dots, b_m$ — позиції веж.

Обмеження

Нехай N і M — суми значень n і m відповідно до всіх тестових випадків.

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $0 \leq f \leq 10^9$
- n - парне
- $1 \leq a_i, b_i \leq 10^9$
- Усі позиції комп'ютерів і веж є різними (у межах одного тестового випадку).

Підзадачі

- Підзадача 1 (5 балів): $N \leq 5000$, $m = 1$
- Підзадача 2 (10 балів): $T \leq 20$, $n \leq 10$, $m \leq 100$
- Підзадача 3 (27 балів): $N, M \leq 5000$

- Підзадача 4 (21 бал): $N \leq 5000$
- Підзадача 5 (37 балів): Без додаткових обмежень.

Вихідні дані

Виведіть T цілих чисел, кожне в окремому рядку — максимально можливу суму оцінок кабелів для кожного тестового випадку.

Приклад

Вхід

```
4
2 1 100
1 10
11
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
```

Вихід

```
89
-4
12
51
```

Vim

Ліміт часу: 1 s Ліміт пам'яті: 256 MiB

Важко знайти щось більш *ліквіське*, ніж старий добрий Vim — редактор, у якому ми можемо робити речі, про які в «звичайних» редакторах можемо лише мріяти. Звісно, тільки якщо ви готові витратити тиждень на вивчення і місяць практики, щоб незвичні, але **потужні** команди, які він підтримує, стали вашою «м'язовою пам'яттю». Саме за це Дмитро його так обоожнює. Отже, як і в інших редакторах, у Vim є *курсор*, який завжди розташований на одному з символів. Спочатку він знаходиться на першому символі тексту, і, як і в інших редакторах, у Vim є *буфер обміну*, призначений для зберігання (а опосередковано — і для копіювання та переміщення) фрагментів тексту. Спочатку буфер обміну порожній.

У цій задачі вважатимемо, що на початку в редакторі є рівно один символ '-' (мінус), а ваша мета — допомогти Дмитру отримати рівно n послідовних символів мінуса. Для цього ви можете використовувати чотири команди:

- **h**: Якщо курсор знаходиться на першому символі, ця команда нічого не робить, інакше переміщує курсор на один символ ліворуч.
- **l**: Якщо курсор знаходиться на останньому символі, ця команда нічого не робить, інакше переміщує курсор на один символ праворуч.
- **Y**: Послідовність символів від позиції курсора до кінця тексту копіюється до буфера обміну, «перезаписуючи» його попередній вміст. (Якщо курсор знаходиться на першому символі тексту, до буфера обміну копіюється весь текст.)
- **P**: Копія тексту, що зберігається в буфері обміну, вставляється перед символом, на якому знаходиться курсор, а курсор переміщується на останній вставлений символ. Вміст буфера обміну не змінюється. Якщо буфер обміну порожній, нічого не відбувається.

Напишіть програму, яка зчитує число n і виводить мінімальну кількість команд, необхідних для отримання рівно n послідовних символів '-' у Vim за описаних умов. Програма також повинна вивести приклад послідовності з мінімальною кількістю команд.

Вхідні дані

Перший рядок містить кількість тестових випадків t . Наступні t рядків містять тестові випадки з бажаною довжиною рядка n .

Обмеження

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

Підзадачі

Якщо ви виведете лише правильну кількість команд, але не приклад, або ж некоректний приклад послідовності команд, то отримаєте лише половину балів за відповідну підзадачу.

- Підзадача 1 (20 балів): $n \leq 100$
- Підзадача 2 (8 балів): $n \leq 1000$
- Підзадача 3 (18 балів): $n \leq 10^4$
- Підзадача 4 (18 балів): $n \leq 10^5$
- Підзадача 5 (18 балів): $n \leq 10^6$
- Підзадача 6 (18 балів): Без додаткових обмежень.

Вихідні дані

Для кожного тестового випадку виведіть необхідну мінімальну кількість команд і приклад такої послідовності команд в окремому рядку.

Приклад

Вхід

2
21
2

Вихід

10 YURhYURPP
2 YP

Примітка

Як видно з наведеної нижче таблиці, у першому випадку послідовність YURhYURPP дає правильний результат. Символ '=' у стовпці Screen позначає символ '=', на якому знаходиться курсор.

Крок	Команда	Екран	Буфер обміну
0		=	(пустий)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	=---	--
5	h	=---	--
6	P	=-----	--
7	Y	=-----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----