

## Зимен лагер

Тази година Иванчо и випуска му решиха да се съберат заедно за зимните празници и да отидат на планина. За да направят престоя си по-забавен, учениците решиха да се разпределят на няколко отбора и да си направят състезание. Като най-отговорен от випуска, на Иванчо получи привилегията първи да избере своя отбор.

За настаняване на отбора си Иванчо избра една поляна с **V** бунгала и **R** пътеки. Всяка пътека свързва две бунгала и винаги може да се стигне от едно бунгало до друго, като се минава само по пътеки. Той ще трябва да реши всеки участник от отбора в кое бунгало да бъде настанен. Във всяко бунгало има място за точно един ученик. Той не е ограничен в това колко хора ще вземе в отбора си, нито е задължен да запълни всички бунгала.

По време на престоя се очаква да има обилен снеговалеж, поради който пътеките между бунгалата ще трябва да се изриват от сняг всяка сутрин. Иванчо знае, че трябва да има достатъчно почистени пътеки, за да може да се минава между всеки две бунгала в които има настанени ученици. Но той също така знае, че съучениците му не са много работливи и  $i$ -тия ученик би помогнал с изриването на най-много **Di** на брой пътеки, започващи от неговото бунгало.

Състезанието ще се проведе под формата на игри, както физически, така и логически. Иванчо знае уменията на съучениците си в различните игри и е поставил оценка на всеки ученик - едно число **Wi**, което показва колко е добър като цяло.

Иванчо нарича съседи двама ученици, които са настанени в бунгала директно свързани с изчистена пътека. Той иска задължително да няма съседи които не са приятели. Не всички ученици са приятели помежду си, но ако настани приятелите в съседни бунгала, това неимоверно би подобрило груповия дух. А всички знаят че колектива е по-силен от индивидуалния състезател. Затова за всеки двама приятели  $i$  и  $j$ , той отбеляза с числото **Cij** колко биха допринесли за екипния дух, ако бъдат настанени в съседни бунгала. ( $Cij = Cji$ ).

За да определи колко добър избор е направил, Иванчо използва следната формула за силата на отбора му:  
**F** = (сумата от  $Cij$  за всички съседи) + (сумата от произведението ( $Wi * (брой\ съседни\ на\ i)$ ), за всеки ученик от отбора на Иванчо).

Той иска да състави отбор с възможно най-висок резултат по горната формула, но не може да се справи сам, затова той се обръща към Вас, добри програмисти. Помогнете на Иванчо, като напишете програма **camp**, която по зададена информация за учениците и бунгалата намира подходящи участници за отбора на Иванчо и тяхното разпределение по бунгала.

### Вход:

На първия ред на входния файл (**camp.in**) се въвеждат две числа **N** и **M**. **N** е броят на учениците, а **M** е броят на двойките приятели.

На следващите **M** реда ще има по три числа - **i**, **j** и **Cij**, където **i** и **j** са номера на ученици, които са приятели, а **Cij** е съответното число за тях.

$$i \neq j;$$

$$0 \leq i, j < N;$$

никоя двойка числа (**i**, **j**) няма да се среща повече от един път;

входът ще е такъв, че учениците да не могат да се разделят на две групи, така че никой от първата да не познава никого от втората.

Следващия ред ще съдържа **N** числа - съответното **Wi** за всеки ученик.

Следващия ред ще съдържа **N** числа - съответното **Di** за всеки ученик.

Следващия ред ще съдържа две числа - **V** и **R**, съответно броя бунгала и броя пътеки на поляната.

Ще следват **R** реда с по две числа - **p**, **q**, които показват, че има пътека между

бунгала  $p$  и  $q$ .

$p \neq q$ ;

$0 \leq p, q < V$ ;

Между всеки две бунгала има най-много една пътека.

**Забележка:** Индексирането на учениците и бунгалата започва от 0.

**Изход:**

На първия ред в изходния файл (**camp.out**) трябва да се изведе едно число  $K$  - броят ученици, които ще участват в отбора на Иванчо.

След това трябва да се изведат  $K$  реда с по две числа  $X_i$  и  $Y_i$  - ученикът с номер  $X_i$  се настанява в бунгало с номер  $Y_i$ .

$0 \leq X_i < N$ ;

$0 \leq Y_i < V$ ;

Едно и също настаняване не трябва да се повтаря в изхода.

На следващия ред трябва да се изведе едно число  $T$  - броят пътеки които трябва да се почистят.

На следващите  $T$  реда трябва да има по две числа  $P_i$  и  $Q_i$  - номерата на учениците, които трябва да изчистят пътеката между бунгалата си.

Трябва да съществува пътека между бунгалата, в които са настанени  $P_i$  и  $Q_i$ .

Една и съща пътека не трябва да се повтаря в изхода.

**Оценяване:**

Ако някое от условията не е изпълнено, програмата получава 0 точки за съответния тестовия пример.

В противен случай дефинираме  $F_{your} = F$ , по формулата, описана по-горе. Нека  $F_{max}$  е максималният резултат, получен от решение на участник за дадения

тестов пример. Тогава програмата ви получава  $\left(\frac{F_{your}+1}{F_{max}+1}\right)^2 \times 100$  процента от точките, предвидени за тестовия пример.

**Ограничения:**

В 15% от тестовете  $N \leq 1000$

В други 30% от тестовете  $N \leq 5000$ .

В останалите тестове  $N \leq 10\ 000$

В 10% от тестовете  $M \leq 10\ 000$

В други 30% от тестовете  $M \leq 50\ 000$ .

В останалите тестове  $M \leq 100\ 000$

В 15% от тестовете  $V \leq 1000$ .

В други 20% от тестовете  $V \leq 2000$ .

В други 35% от тестовете  $V \leq 5000$ .

В останалите тестове  $V \leq 10\ 000$ ;

В 30% от тестовете  $R \leq 10\ 000$ .

В други 30% от тестовете  $R \leq 20\ 000$ .

В други 25% от тестовете  $R \leq 50\ 000$ .

В останалите тестове  $R \leq 100\ 000$ ;

$0 \leq C_{ij} \leq 1000$

$0 \leq W_i \leq 100$

Нека  $G_1$  е графът, образуван с върхове учениците и ребра приятелствата между тях. Нека  $G_2$  е графът, образуван с върхове бунгалата и ребра пътеките между тях.

В 15% от тестовете,  $G_1$  и  $G_2$  са дървета.

В други 25% от тестовете, само  $G_1$  е дърво.

В тестовете графите  $G_1$  и  $G_2$  са генерирани така, че поне между 20% и 60% от

структурата им съвпада. Останалата част от графа е генерирана случайно.

**Ограничение за време:** 5 сек.

**Пример 1:**

camp.in	camp.out
6 10	6
0 1 2	2 0
0 2 4	5 1
0 5 7	3 2
1 3 8	1 3
1 5 11	4 4
2 4 12	0 5
2 5 13	6
3 4 16	0 2
3 5 17	2 4
4 5 19	4 3
10 5 2 1 3 0	3 1
3 3 3 3 3 3	1 5
6 6	5 0
0 4	
0 5	
1 3	
1 5	
2 3	
2 4	
F <sub>your</sub> = 100	

Обяснение:

$$F = (4 + 12 + 16 + 8 + 11 + 7) + (2 \cdot 3 + 2 \cdot 0 + 2 \cdot 2 + 2 \cdot 5 + 2 \cdot 1 + 2 \cdot 10) = 100$$

Да се отбележи, че това решение не дава оптимален резултат.

**Пример 2:**

camp.in	camp.out
6 10	5
0 1 2	1 0
0 2 4	0 1
0 5 7	2 2
1 3 8	4 3
1 5 11	5 5
2 4 12	4
2 5 13	1 0
3 4 16	0 2
3 5 17	0 5
4 5 19	5 4
10 5 2 1 3 0	
3 1 1 0 1 2	
6 7	
0 1	
0 5	
1 2	
1 5	
2 3	
3 4	
3 5	
F = 72	