

Финал. Полети

Не далеч в нашата галактика се намира планетарната система Ителоп, която е известна с множеството си от населени планети. Както се досещате, обитателите ѝ имат нужда, както да могат да стигат от един град до друг, така и от една планета до друга. Затова в Ителоп е силно развит бизнесът с междупланетарни пътнически полети. В началото на годината всички компании, които предоставят транспортни услуги, се опитват да изработят такъв график на полетите, че да задоволят колкото може по-добре нуждите на пътниците и да максимизират своята печалба. Измежду всичките сто програмисти, които кандидатстваха за да помогнат в решаването на тази не лека задача, бяха избрани 12, един от които сте и вие.

Обитаваните планети в системата Ителоп са N , номерирани от 1 до N . Компаниите разполагат с M космически кораба, номерирани от 1 до M . Както знаете, различният размер и позиция на планетите означава за различните планети – различен брой часове в денонощието, различен брой дни в месеца и различен брой месеци в годината. Затова всички планети в системата използват времеизмерването на главната планета α -Ителоп – ден от 16 часа, месец от 64 дни и година от 4 месеца.

В началния момент, i -тият кораб се намира на планетата с номер B_i , а характеристиките му са – скорост S_i километра в час, цена – C_i Ителопски лева за километър и капацитет – Cap_i пътника. Поради сложната поддръжка на корабите са необходими два вида диагностика. За първия е необходимо всеки кораб да се върне на планетата, от която е излетял, след като са изминали не повече от четири дни (64 часа) от последната диагностика. Самата диагностика започва в 00:00 часа на следващия ден и трае 16 часа, през което време корабът не може да лети, след което отново може да поеме полети. Т.е. ако корабът се върне на планетата, от която е излетял първоначално, в 10:00 часа на третия ден, то най-рано може да започне отново да прави полети след втората диагностика на петия. Ако някой кораб не успее да се завърне в рамките на 64 часа на родната планета, на компанията се отнема лиценз за провеждане на полети, за това и такъв отговор не е допустим (ще носи 0 точки). Втората диагностика се провежда всеки ден от 00:00 до 04:00 по часовника на α -Ителоп. Тогава, разбира се, корабът няма право да лети, но пък за тази профилактика не е нужно да се връща на родната планета.

След като някой от корабите направи определен брой полети, върне се на родната си планета и му бъде направена първият вид диагностика, в същия начален час на следващия ден той започва да изпълнява същия маршрут (ще го наричаме **цикъл** на кораба), докато отново се върне на родната планета и т.н. и т.н. ще повтаря цикъла си до края на годината (не е задължително в края на година корабите да са на родните си планети).

Поради политически конфликти, може да се лети само между зададени F двойки планети A_j и B_j , като е зададено и разстоянието D_j между тях. Гарантирано е, че щом може да се лети от A_j до B_j то ще може да се лети и от B_j до A_j , но те ще бъдат зададени като друга двойка и разстоянието в този случай може да е различно от D_j .

Очакваният поток пътници през различните месеци на годината е различен, но за всеки ден от един месец се очаква един и същ поток. За всяка двойка планети A_j и B_j , между които е позволено да се лети, ще има точно K_j извършени полета. Един полет се дефинира с начален час SH_k , краен час EH_k , месец O_k , очаквана печалба от един пътник CT_k и очакван брой пътници P_k , които трябва да бъдат превозени. Това означава, че за всеки ден от месец O_k , в часовете от SH_k до EH_k може да бъде извършен полет, който да превози P_k пътника от планета A_j до планета B_j и това ще донесе печалба CT_k . Ако капацитетът на един кораб е по-малък от очаквания брой на пътниците или часът, в който ще пристигне, е по-късен от EH_k – корабът няма право да изпълни полета. Забележете, че ако пристигне по-рано от EH_k това не е проблем. Излитането обаче става единствено в кръгли часове т.е., ако корабът пристигне в 5 часа и 34 минути, най-ранният момент, в който може да излети, е 06:00.

Корабите имат право да летят между позволените планети дори и за съответния времеви интервал да няма пътници. В такъв случай обаче няма да имат никаква печалба, а само разходи. Два кораба нямат право да летят между една и съща двойка планети, излитайки в един и същи ден и час, движейки се в една и съща посока.

Помогнете на компаниите. Напишете програма, която да състави разписанието на полетите, което ще донесе максимална печалба, като за всеки кораб определи **цикъла**, който той трябва да извърши. Всеки цикъл трябва да е с положителна дължина, независимо от това, че максималната печалба може да е отрицателна.

Оценяване:

Оценяването за един тест ще стане по формулата $\frac{|\text{yours} - \text{min}| + 1}{|\text{max} - \text{min}| + 1} \times \frac{100}{T}$, където **yours** е максималната цена, която програмата ви намери, **min** е минималната, а **max** – максималната цена получени от някой състезател за този тест, а T е броят тестове.

Вход:

На първия ред от входния файл **flights.in** се задават числата N , M и F , съответно броят планети, броят кораби и броят позволени двойки планети, между които може да се лети.

Следват M реда с по четири числа B_i , S_i , C_i и Cap_i на всеки – стартовата планета, скоростта, цената за километър и капацитета на i -тия кораб. След това, за всяка от F -те двойки ще бъдат зададени числата A_j , B_j , D_j и K_j . Всяка такава четворка ще бъде последвана от K_j реда, всеки съдържащ числата SH_k , EH_k , O_k , CT_k и P_k описващи k -тия полет между двете планети.

Изход:

В изходния файл **flights.out** програмата трябва да изведе M реда, на всеки от които ще има описание на цикъла, който трябва да прави съответния кораб. Описанието на цикъл започва с броя Y_m на полетите, които прави корабът в един цикъл. Следват Y_m тройки числа – денят, в който

ще се състои полета в рамките на цикъла (число от 1 до 4), час на излитане на кораба и номерът на планетата, на която ще кацне.

Ограничения:

$1 < N \leq 50$

$1 < M \leq 50$

$0 < F \leq 1000$

$0 < C_i < 256$

$0 < S_i, P_k, Cap_i, CT_k, D_j < 512$

Времето за изпълнение на програмата е 2 секунди.

Пример:

flights.in	flights.out	Checker message:
5 2 8 1 10 5 15 3 10 2 15 2 1 2 0 1 2 2 2 5 10 1 20 5 5 10 4 20 5 3 2 2 0 2 3 2 3 5 10 1 20 6 5 10 3 20 6 5 10 4 20 6 4 3 2 0 3 4 2 1 5 10 1 10 10 1 4 2 0 4 1 2 1 5 10 4 10 1	4 1 4 2 2 4 3 3 4 4 4 4 1 3 1 5 4 2 5 1 3 5 2	A spaceship didn't manage to get back to the base for maintenance. Wrong answer!
5 2 8 1 10 5 15 3 10 2 15 2 1 2 0 1 2 2 2 5 10 1 20 5 5 10 4 20 5 3 2 2 0 2 3 2 3 5 10 1 20 6 5 10 3 20 6 5 10 4 20 6	4 1 4 2 2 4 3 3 4 4 4 4 1 4 1 5 4 2 5 1 <u>4 5 2</u> <u>4 10 3</u>	OK: 1150

4 3 2 0		
3 4 2 1		
5 10 1 10 10		
1 4 2 0		
4 1 2 1		
5 10 4 10 1		
5 2 8	4	Two spaceships should not fly from one planet to another in the same time and direction. Wrong answer!
1 10 5 15	1 4 2	
<u>1 10 2 15</u>	2 4 3	
2 1 2 0	3 4 4	
1 2 2 2	4 4 1	
5 10 1 20 5	4	
5 10 4 20 5	1 4 4	
3 2 2 0	2 4 3	
2 3 2 3	3 4 4	
5 10 1 20 6	4 4 1	
5 10 3 20 6		
5 10 4 20 6		
4 3 2 0		
3 4 2 1		
5 10 1 10 10		
1 4 2 0		
4 1 2 1		
5 10 4 10 1		

Забележка: не е задължително отговорът на втория пример да е оптималният. Възможно е да съществува по-добър.

Генериране на тестовете:

Финалните тестове ще бъдат генерирани в четири групи:

№	%	N	M	F	make_random_graph(G, F) ще връща
1	15	25	25	500	произволен свързан граф
2	15	25	25	500	произволен граф
3	15	50	50	1000	произволен граф
4	55	50	50	1000	произволен свързан граф

Забележка: това, че графът е произволен не означава, че не е свързан.

make_random_graph(G, F) е функцията от тест генератора на следващата страница. За повече информация как е имплементирана, можете да видите в изходния код на генератора, който се намира в сайта на състезанието.

```
set_random_seed(seed);

println(N, M, F);

for i from 1 to M:
    base := rand(1, N);
    speed := rand(16, 512);
    cost := rand(16, 256);
    capacity := rand(16, 512);

    println(base, speed, cost, capacity);

make_random_graph(G, F); //create random graph G with N nodes and F edges

foreach edge A->B in G:
    dist(A, B) := rand(64, 512);
    K := rand(0, 16);

    println(A, B, dist(A, B), K);

    for k from 1 to K:
        month := rand(1, 4);
        start_hour := rand(4, 15);
        end_hour := rand(start_hour, 16);

        if ([A][B][month][start_hour] is already used):
            try again for the same k;

        mark [A][B][month][start_hour] as used;

        cost := rand(1, 512);
        passengers := rand(16, 512);

    println (start_hour, end_hour, month, cost, passengers);
```

Ако искате да използвате генератора, с който са направени финалните тестове, е необходимо да го стартирате, подавайки му два параметъра – число от 1 до 4 съответстващо на това от коя група да бъде теста и число от 0 до 2^{31} , задаващо съответния **seed**, с който да се инициализира генератора на произволни числа. Следният пример ще извика генератора, при което той ще направи файл **flights.out**, в който ще има тест от четвъртата група генериран със **seed = 666**:

```
C:\> test_generator.exe 4 666
```
