Algorytmy i struktury danych, PS, studia stacjonarne, zestaw zadań nr 5,
Krzysztof Ostrowski

Problemy grafowe

Rozwiązanie każdego z zadań polega na:

1) **zaproponowaniu możliwie najefektywniejszego czasowo rozwiązania**
2) **określeniu jego pesymistycznej złożoności czasowej**

Problem 1 - „Misja”

Jest XXXIII wiek. Ludzie podróżują ultra-szybkimi statkami kosmicznymi pomiędzy zawieszonymi w przestrzeni międzygwiezdnej ogromnymi hipermiastami. Kosmos jest w stanie wojny i dowolne podróżowanie w przestrzeni kosmicznej jest zabronione. Do komunikacji służą tylko specjalnie wybrane korytarze komunikacyjne łączące ze sobą hipermiasta. W tych korytarzach statki kosmiczne poruszają się z prędkością światła, jednak odległości między hipermiastami są duże – liczone zwykle w godzinach świetlnych. Stąd podróże trwają pewien czas. Niektóre hipermiasta są jednak połączone specjalnymi tunelami czasoprzestrzennymi i dzięki temu podróż między nimi nic nie trwa (!). Takich tuneli w całej przestrzeni jest jednak co najwyżej 50. Z racji tego, że trwa kosmiczna wojna, to działania szpiegowskie są zakrojone na dużą skalę. Pewien szpieg znajduje się w pierwszym hipermieście i ma za zadanie wykraść tajne informacje, które są ukryte w ostatnim z hipermiast. W związku z tym musi przemieścić się on z hipermiasta nr 1 do hipermiasta nr *n*. Po drodze może on swobodnie poruszać się zwykłymi korytarzami. Jednak podróżowanie tunelami czasoprzestrzennymi jest niebezpieczne ze względu na ich niestabilność – szpieg może po prostu zniknąć w wyższym wymiarze. Nie da się jednak ukryć, że tunele mogą pomóc w szybszym pokonywaniu trasy. Szpieg stosuje w związku z tym zasadę ograniczania ryzyka i w czasie całej podróży pozwala sobie na skorzystanie z co najwyżej *k* różnych tuneli czasoprzestrzennych (z każdego po razie). Misja jest krytyczna i od jej powodzenia mogą zależeć losy wszechświata. Szpieg musi ją wykonać w jak najkrótszym czasie. Czy pomożesz mu w wykonaniu zadania i znalezienia najkrótszej czasowo trasy?

Wejście:

W pierwszej linii wejścia podana jest liczba n hipermiast, m korytarzy oraz maksymalną liczbę k tuneli czasoprzestrzennych, których można użyć (1<*n*<=10000, 1<=*m*<=100000, 1<=*k*<=3). W kolejnych *m* liniach są podane informacje dotyczące korytarzy: po 3 liczby całkowite oznaczające odpowiednio numery połączonych miast oraz długość korytarza (co najwyżej 100 godzin świetlnych).

Wyjście:

W pierwszej linii wyjście należy podać czas optymalnej trasy, a w kolejnej linii po kolei ciąg liczb (numerów miast) opisujących trasę.

Przykład:

Wejście:	Wyjście:
5 7 1	4
1 3 3 //między 1 i 3 korytarz powietrzny o długości 3 godzin świetl.	1 2 4 5
2 4 2 //itd.	
2 3 3	
1 2 0 //tunel czasoprzestrzenny	
3 4 0 //tunel czasoprzestrzenny	
3 5 3	
4 5 2	

Problem 2 - „Rzym”

Cezar rządził w starożytnym Rzymie i budował dwukierunkowe trakty, by ułatwić podróżowanie. Sieć połączeń między poszczególnymi miastami gęstniała z roku na rok, czym chlubili się Rzymianie. Przed budową nowego odcinka drogi Cezar zawsze konsultował się ze swoimi doradcami. W miarę jak dróg przybywało, coraz trudniej było zdecydować, w którym miejscu najlepiej zbudować nowy trakt. Narady trwały długo, a Cezar wyraźnie się niecierpliwił. Priorytetem Cezara przy budowie nowego traktu było to, by jak najwięcej miast i miasteczek miało dzięki temu ułatwiony dojazd do stolicy. Wszak wszystkie drogi prowadzą do Rzymu (a przynajmniej powinny). Potrzeba tęgiej głowy, by rozstrzygnąć, czy budowa takiego lub innego traktu skróci dojazd do stolicy Imperium. Zadanie jest trudne, a gniew Cezara zaczyna rosnąć...
O wiele łatwiej by było mieć do dyspozycji komputer. Ty go masz i to Ty odpowiesz na pytania stawiane przez Cezara.

Wejście:
W pierwszej linii wejścia podane są dwie liczby całkowite *n*, *m* i *k* (1<=*n*<=50000, 1<=*m*<=500000, 1<=*k*<=1000) oznaczające liczbę miast, liczbę wybudowanych traktów oraz liczbę kandydatów na trakty. W kolejnych m liniach znajdują się dane o wybudowanych już traktach: w każdej po 3 liczby całkowite *m1*, *m2*, *d* (1<=*m1*, *m2*<=*n*, 1<=*d*<=1000) oznaczające odpowiednio miasta połączone traktem i długość tego traktu. W kolejnych d liniach znajdują się dane potencjalnych traktów.

Wyjście:
W kolejnych *d* liniach wyjścia należy wypisać, ile miast skróciłoby swą drogę do stolicy (miasta nr 1) gdybyśmy wybudowali dany trakt. Jeśli liczba tych miast przekracza 100 należy napisać '100+'.

Przykład:
Wejście:
```
6 6 3        //6 miast, 6 wybudowanych traktów, 3 potencjalne trakty
1 2 2        //trakt między 1 i 2 miastem ma długość 2
1 3 5        //itd.
3 5 1
2 4 6
4 6 5
5 6 4
1 6 8        //potencjalny trakt między miastami 1 i 6 o długości 8
2 5 1        //itd
3 4 3
```

Wyjście:
```
1    //wybudowanie pierwszego potencjalnego traktu skróciłoby drogę do stolicy z jednego miasta
3    //itd.
0
```

Problem 3 - „Koleje”

W pewnym postępowym kraju od stuleci rozwija się transport kolejowy. Początkowo połączenia kolejowe obejmowały tylko największe i najszybciej rozwijające się miasta, a prędkość pociągów pozostawiała wiele do życzenia. Według opowieści starszych można było wyskoczyć na grzyby, gdy pociąg przejeżdżał przez las, by po grzybobraniu załapać się jeszcze do ostatniego wagonu. Z czasem prędkości pociągów rosły, by w końcu doprowadzić do likwidacji konwencjonalnych przejazdów kolejowych (ze względów bezpieczeństwa). Zmienił się też wygląd pociągów – początkowo przypominały fabryki z kominem, a obecnie bliżej im do filmów SF. W kraju tym nie toleruje się obniżenia prędkości pociągów i po modernizacji danego odcinka średnia prędkość na nim może jedynie wzrosnąć. Co więcej, postój na każdej stacji wynosi 5 minut i nigdy nie ma opóźnień, a z żadnego miasta nie wychodzi więcej niż 50 połączeń. W bazie danych kolei istnieje całe archiwum dotyczące budowy i modernizacji odcinków łączących poszczególne miasta. Zlecono napisanie oprogramowania do analizy tych danych. Twoim zadaniem jest ustalenie, kiedy czas przejazdu pociągiem między zadaną parą miast skrócił się do zadanego poziomu.

Wejście:
W pierwszej linii wejścia podane są trzy liczby całkowite n, m i z (1<=n<=10000,1<=m<=100000, 1<=z<=10) oznaczające odpowiednio liczbę miast, liczbę budowanych/modernizowanych odcinków oraz liczbę zapytań do programu. W kolejnych m liniach znajdują się chronologiczne informacje dotyczące budowy. W każdej z tych linii podane są: data (format rrrr-mm-dd), następnie znak 'm' lub 'b' w zależności od tego czy chodzi o budowę (połączenie wcześniej nie istniało) czy o modernizację istniejącego już odcinka, następnie para różnych miast $m1$, $m2$ (1<=$m1$, $m2$<=n) między którymi budowane/modernizowane jest połączenie, a następnie średnią prędkość v (w km/h) na tym odcinku (1<=v<=500) oraz (w przypadku budowy) również długość d (w km) nowo-budowanego połączenia (1<=d<=1000). v i d są liczbami całkowitymi i $60*d \bmod v = 0$. W kolejnych z liniach znajdują się zapytania złożone z liczb $m1$, $m2$, c (1<=$m1$, $m2$<=n, 1<=c<=10000) oznaczających parę miast oraz maks. czas połączenia między nimi (w minutach).

Wyjście:
W z liniach wyjścia należy odpowiedzieć na zapytania: kiedy po raz pierwszy udało się osiągnąć czas najszybszego połączenia (pośredniego lub bezpośredniego) między danymi miastami nie przekraczający zadanego limitu. Jeśli taki czas nie został dotąd osiągnięty, należy wypisać 'NIE'.

Przykład:

Wejście:
```
5 10 3                      //5 miast, 10 połączeń, 3 zapytania
1900-05-30 b 1 2 30 60      //budowane połączenie między miastami 1 i 2, v=30 km/h, d=60 km
1900-07-15 b 2 5 40 120     //itd.
1905-04-19 b 1 4 35 70
1910-06-03 b 4 5 50 100
1950-10-25 m 2 5 72         //modernizowane połączenie między miastami 2 i 5: nowe v=72 km/h
1955-01-31 b 1 3 60 90
1990-12-12 b 3 5 180 90
2000-09-20 b 3 4 240 60
2005-06-14 m 1 4 280        //itd.
2008-03-07 b 2 3 150 50
1 5 230                //Kiedy pomiędzy miastami 1 i 5 osiągnięto czas podróży maks. 230 minut
1 5 75                 //itd.
2 4 35
```
Wyjście:
```
1950-10-25   //od 1950-10-25 między miastami 1 i 5 dało się przejechać w  225 minut (<=230)
2005-06-14   //od 2005-06-14 między miastami 1 i 5 dało się przejechać w 70 minut (<=75)
NIE          //najszybsze połączenie między 2 i 4 wymaga obecnie 40 minut
```