Karol Pokorski
Treść zadania, Opracowanie

Marcin Andrychowicz
Program

Dostępna pamięć: 128 MB. *OI, etap III, dzień drugi, 17.04.2015*

Wycieczki

Bajtazar odkrył piękno wycieczek rowerowych. Swój k-dniowy urlop zamierza spędzić w przepięknej Bajtocji. Każdego dnia chce wybrać się na wycieczkę rowerową biegnącą inną trasą. Chciałby stopniować poziom trudności wycieczek, zatem każda kolejna wycieczka nie powinna być krótsza od poprzedniej. A dokładniej: i-tego dnia Bajtazar chce zrobić wycieczkę i-tą najkrótszą możliwą trasą biegnącą przez miasta Bajtocji. Pomóż Bajtazarowi obliczyć długość ostatniej wycieczki, na którą wybierze się k-tego dnia urlopu.

W Bajtocji znajduje się n miast, ponumerowanych od 1 do n. Miasta są połączone jednokierunkowymi drogami, a długość każdej drogi wynosi 1, 2 lub 3 kilometry. Drogi mogą prowadzić tunelami i estakadami. Rozważamy wycieczki zaczynające się i kończące w dowolnych miastach Bajtocji. Dopuszczamy też wycieczki prowadzące przez dane miasto lub drogę wielokrotnie.

Wejście

Pierwszy wiersz standardowego wejścia zawiera trzy liczby całkowite n, m i k ($1 \leqslant n \leqslant 40$, $1 \leqslant m \leqslant 1000$, $1 \leqslant k \leqslant 10^{18}$) pooddzielane pojedynczymi odstępami, oznaczające liczbę miast, liczbę dróg i liczbę dni urlopu. W kolejnych m wierszach znajdują się opisy dróg, po jednej w wierszu. Opis każdej drogi składa się z trzech liczb całkowitych u, v i c ($1 \leqslant u, v \leqslant n$, $u \neq v$, $1 \leqslant c \leqslant 3$) pooddzielanych pojedynczymi odstępami, oznaczających jednokierunkową drogę z miasta u do miasta v o długości c kilometrów. Pomiędzy parą miast może być więcej niż jedna droga.

W testach wartych łącznie 75% punktów zachodzą dodatkowo warunki $n \leqslant 15$, $m \leqslant 200$, $k \leqslant 10^{12}$. Ponadto w podzbiorze tych testów wartym łącznie 50% punktów zachodzi dodatkowo dla każdej drogi $c = 1$. W końcu w podzbiorze tych testów wartym łącznie 25% punktów zachodzi jeszcze dodatkowo warunek $k \leqslant 1\,000\,000$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia Twój program powinien wypisać długość k-tej najkrótszej wycieczki. Jeśli jest mniej niż k różnych wycieczek (czyli Bajtazar będzie zmuszony zakończyć swój urlop wcześniej), należy wypisać -1.

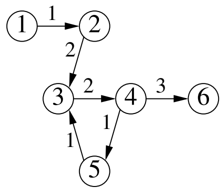
Przykład

Dla danych wejściowych:

```
6 6 11
1 2 1
2 3 2
3 4 2
4 5 1
5 3 1
4 6 3
```

poprawnym wynikiem jest:

```
4
```



Wyjaśnienie do przykładu:

Wycieczki długości 1: $1 \to 2$, $5 \to 3$, $4 \to 5$.
Wycieczki długości 2: $2 \to 3$, $3 \to 4$, $4 \to 5 \to 3$.
Wycieczki długości 3: $4 \to 6$, $1 \to 2 \to 3$, $3 \to 4 \to 5$, $5 \to 3 \to 4$.
Jedenastą najkrótszą wycieczką (długości 4) może być na przykład: $5 \to 3 \to 4 \to 5$.

Testy „ocen”:

`1ocen`: $n = 10$, $k = 46$; *drogi o losowych długościach tworzące ścieżkę; istnieje tylko 45 możliwych wycieczek, więc poprawną odpowiedzią jest* -1;

`2ocen`: $n = 15$, $k = 10^{12}$; *z każdego miasta do każdego innego jest droga o długości 3.*

Rozwiązanie

W zadaniu dany jest niewielki skierowany graf G zawierający n wierzchołków ($n \leqslant 40$). W grafie mogą występować krawędzie wielokrotne. Waga każdej krawędzi należy do zbioru $\{1, 2, 3\}$. Należy wyznaczyć długość k-tej najkrótszej ścieżki w tym grafie.

Rozwiązanie dla jednostkowych wag krawędzi

Spróbujmy najpierw rozwiązać łatwiejszą wersję zadania, w której wagi wszystkich krawędzi są równe 1.

Graf podany na wejściu można przedstawić w postaci macierzy sąsiedztwa M, w której wyraz $M_{u,v}$ będzie równy liczbie krawędzi prowadzących z wierzchołka u do wierzchołka v.

Mnożenie macierzy

Powiemy, że macierz M jest wymiaru $p \times q$, jeżeli zawiera p wierszy i q kolumn. Reprezentowanie macierzy w pamięci komputera jest bardzo łatwe i sprowadza się do stworzenia tablicy dwuwymiarowej. Co ważniejsze jednak, do rozwiązania zadania będziemy używać własności macierzy jako pojęcia matematycznego z algebry liniowej, nie zaś jedynie jako reprezentacja informacji w pamięci komputera.

Jedną z podstawowych operacji dotyczących macierzy jest możliwość ich mnożenia. W wyniku pomnożenia macierzy A wymiaru $p \times q$ i macierzy B wymiaru $q \times r$ uzyskujemy macierz $A \cdot B$ wymiaru $p \times r$, w której każdy wyraz możemy obliczyć następującym wzorem:

$$(A \cdot B)_{u,v} = \sum_{w=1}^{q} A_{u,w} \cdot B_{w,v}. \tag{1}$$

Operacja mnożenia macierzy, w przeciwieństwie do mnożenia liczb, nie jest operacją przemienną, jest jednak operacją łączną. Dodatkowo warty odnotowania jest fakt, że macierze A i B nie muszą być kwadratowe, aby móc je pomnożyć, jednak powyższy wzór (i definicja operacji mnożenia macierzy) wymusza, żeby liczba kolumn macierzy A była równa liczbie wierszy macierzy B. Do rozwiązania zadania będziemy używać jedynie macierzy kwadratowych, co dodatkowo ułatwia analizę problemu (oraz implementację rozwiązania).

Obliczenie iloczynu dwóch macierzy wymiaru $p \times p$ można wykonać bezpośrednio ze wzoru (1) w czasie $O(p^3)$, gdyż obliczenie każdego z p^2 wyrazów macierzy wynikowej zajmuje czas liniowy względem p.

Istnieją efektywniejsze metody mnożenia macierzy, na przykład algorytm Strassena ($O(p^{\log_2 7})$) lub algorytm Coppersmitha-Winograda ($O(p^{2,38})$). Zastosowanie tych algorytmów niekoniecznie jednak jest użyteczne na zawodach, ze względu na trudność implementacji oraz stosunkowo duży stały czynnik ukryty w notacji asymptotycznej.

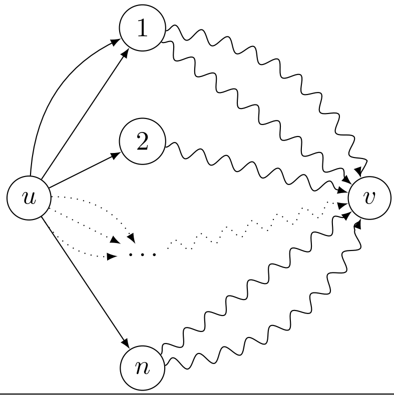
Potęgowanie macierzy a ścieżki w grafie

Oprócz mnożenia macierzy można zdefiniować operację potęgowania macierzy M (tym razem już tylko kwadratowej). Dla potęg całkowitych dodatnich będzie ona określona podobnie do mnożenia liczb: $M^1 = M$, $M^{t+1} = M \cdot M^t$.

Okazuje się, że operacja potęgowania macierzy ma bardzo silne powiązanie z wyznaczaniem liczby ścieżek w grafie. Dokładniej, wyraz w u-tym wierszu i v-tej kolumnie t-tej potęgi macierzy sąsiedztwa M grafu G jest równy liczbie różnych ścieżek t-krawędziowych z wierzchołka u do wierzchołka v.

Fakt ten możemy udowodnić, korzystając z indukcji matematycznej: dla $t = 1$ każdy wyraz macierzy $M_{u,v}$ jest po prostu liczbą krawędzi (czyli ścieżek składających się z jednej krawędzi) pomiędzy u i v. Dowolną ścieżkę $(t+1)$-krawędziową (dla $t \geqslant 1$) pomiędzy u i v możemy zawsze podzielić na dwie części: część pierwszą (nazwijmy ją *krótką*) jednokrawędziową z wierzchołka u do pewnego wierzchołka pośredniego w oraz część drugą (nazwijmy ją *długą*) t-krawędziową z wierzchołka w do v (rys. 1). Aby zliczyć wszystkie ścieżki $(t+1)$-krawędziowe pomiędzy wierzchołkami u i v wystarczy zatem zsumować liczbę takich ścieżek dla każdego wierzchołka pośredniego w. Dla ustalonego wierzchołka pośredniego w dowolna krótka ścieżka z u do w (jest ich $M_{u,w}$) z dowolną długą ścieżką z w do v (na mocy założenia indukcyjnego jest ich $(M^t)_{w,v}$) tworzą poprawną $(t+1)$-krawędziową ścieżkę z u do v. Zatem liczba tych ścieżek to:

$$\sum_{w=1}^{n} M_{u,w} \cdot (M^t)_{w,v} = (M \cdot M^t)_{u,v} = (M^{t+1})_{u,v}.$$



Rys. 1: Ilustracja dowodu kroku indukcyjnego: ścieżki $(t+1)$-krawędziowe z wierzchołka u do wierzchołka v. Krótkie krawędzie reprezentują ścieżki jednokrawędziowe (część krótką), zaś krawędzie falowane reprezentują ścieżki t-krawędziowe (część długą).

Bezpośredni algorytm obliczania t-tej potęgi macierzy wymiaru $n \times n$ działa w czasie $O(n^3 \cdot t)$. Jednak dzięki temu, że operacja mnożenia macierzy jest łączna, można zastosować tę samą technikę co w szybkim potęgowaniu liczb: $M^{2t} = M^t \cdot M^t$ oraz $M^{2t+1} = M \cdot M^t \cdot M^t$. W ten sposób można zredukować wykładnik potęgi dwukrotnie, wykonując maksymalnie dwa mnożenia macierzy, co prowadzi do redukcji liczby mnożeń z liniowej do logarytmicznej względem t i złożoności czasowej $O(n^3 \cdot \log t)$. Można na to też spojrzeć jak na wykonanie innego podziału ścieżki na część krótką i długą – wykonując bardziej zrównoważony podział, można sprowadzić większy problem (obliczenia potęgi t macierzy M) do dwóch takich samych, ale sporo mniejszych problemów (obliczenia potęgi $\lfloor \frac{t}{2} \rfloor$ macierzy M).

Zliczanie ścieżek długości co najwyżej ℓ

Na podstawie rozumowania z potęgowaniem macierzy można wyznaczyć liczbę ścieżek długości równej ℓ (po prostu licząc M^ℓ i sumując wszystkie wyrazy otrzymanej macierzy).

Co jednak, jeśli chcielibyśmy wyznaczyć liczbę ścieżek o długości co najwyżej ℓ? Można stworzyć graf G' poprzez dodanie do grafu G dodatkowego wierzchołka startowego s, z którego wychodzą pojedyncze krawędzie jednostkowej długości do wszystkich pozostałych wierzchołków z grafu G, a także pętla do wierzchołka s. Wówczas liczba ścieżek długości dokładnie $\ell + 1$ z s do wszystkich pozostałych wierzchołków (poza s) w grafie G' jest równa liczbie ścieżek długości co najwyżej ℓ w oryginalnym grafie G. Faktycznie, ścieżka $u \rightsquigarrow v$ długości $t \leqslant \ell$ w grafie G odpowiada ścieżce długości $\ell + 1$ w grafie G', w której najpierw $\ell - t$ razy przechodzimy pętlą przy wierzchołku s, potem krawędzią z s do u i na końcu ścieżką $u \rightsquigarrow v$.

Właściwe rozwiązanie

Wróćmy do zadania wyznaczenia k-tej najkrótszej ścieżki w grafie G. Wiadomo, że jeśli rozwiązanie istnieje, to musi mieć długość co najwyżej k. Gdyby tak nie było, czyli gdyby szukana ścieżka miała długość większą niż k, to rozpatrując wszystkie prefiksy tejże ścieżki, uzyskalibyśmy co najmniej k ścieżek od niej krótszych – sprzeczność.

A zatem skoro wiadomo, że długość szukanej ścieżki (wykładnik potęgi macierzy M) musi być równa co najmniej 1 oraz co najwyżej k, rozwiązanie zadania nasuwa się samo: możemy zastosować wyszukiwanie binarne długości ścieżki. Należy znaleźć takie ℓ, że ścieżek długości mniejszej niż ℓ będzie mniej niż k, zaś ścieżek długości co najwyżej ℓ będzie już co najmniej k. W ten sposób uzyskaliśmy rozwiązanie uproszczonej wersji zadania (dla wag jednostkowych krawędzi) działające w czasie $O(n^3 \log^2 k)$. Zgodnie z uwagą w treści zadania takie rozwiązanie warte było na zawodach 50% punktów. Implementację tego rozwiązania Czytelnik znajdzie w pliku `wycb1.cpp`.

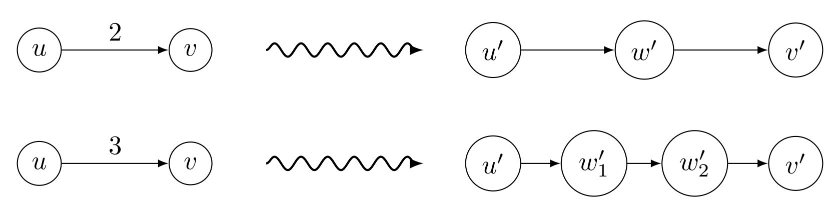
Rozwiązanie to można nieco poprawić. Zamiast wyszukiwania binarnego można najpierw obliczyć liczby ścieżek o długościach co najwyżej 1, 2, 4, 8, ... (kolejne potęgi dwójki), aż liczba ścieżek przekroczy k. W ten sposób obliczymy co najwyżej $\log k$ macierzy: M, M^2, M^4, M^8, ... w czasie $O(n^3 \log k)$, a także ustalimy pozycję najstarszego zapalonego bitu wyniku (długości szukanej ścieżki), czyli znajdziemy takie M^{2^p}, które daje co najwyżej k ścieżek, że $M^{2^{p+1}}$ daje więcej niż k ścieżek. Teraz można próbować ustalać kolejne bity wyniku (od lewej do prawej) poprzez domnażanie do ostatniej uzyskanej macierzy (M^{2^p}) wcześniej obliczonych macierzy dla coraz mniejszych wykładników ($M^{2^{p-1}}$, $M^{2^{p-2}}$, ...).

Zobrazujmy powyższy pomysł na krótkim przykładzie, w którym poszukiwany wynik wynosi $\ell = 11$. Najpierw, obliczając kolejno M^1, M^2, M^4, M^8, algorytm uzyskuje nie więcej niż k ścieżek, zaś w M^{16} jest już ich więcej niż k. To znaczy, że wynik jest pomiędzy 8 a 15 (ma cztery bity). Do macierzy M^8 można domnożyć uprzednio obliczoną M^4, co da w wyniku M^{12}, której suma liczby ścieżek będzie zbyt duża (większa niż k), więc wynik musi być pomiędzy 8 a 11 (drugi najstarszy bit jest zgaszony). Następnie do macierzy M^8 można domnożyć uprzednio obliczoną M^2, co da M^{10} i mniej niż k ścieżek, czyli trzeci bit wyniku jest równy 1. Na końcu do macierzy M^{10} algorytm domnaża M^1 i sprawdza, że wynikiem jest $\ell = 11$.

Łącznie mnożeń macierzy w tej fazie będzie też $O(\log k)$, co powoduje, że złożoność całego rozwiązania redukuje się do $O(n^3 \log k)$.

Rozwiązanie dla krawędzi o wagach $\{1, 2, 3\}$

W macierzy sąsiedztwa M dla grafu G zapamiętujemy jedynie liczbę krawędzi pomiędzy wierzchołkami. Na przechowanie wag krawędzi nie ma już miejsca. Teoretycznie można by rozważyć dodanie dodatkowych wierzchołków pomiędzy końcami krawędzi o wagach 2 i 3; patrz rys. 2.



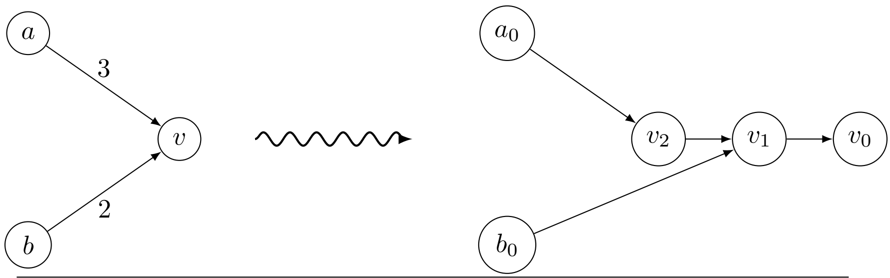
Rys. 2: Dodatkowe wierzchołki pomiędzy końcami krawędzi.

Niestety, taka metoda w pesymistycznym przypadku powoduje zwiększenie liczby wierzchołków o dwukrotność liczby krawędzi w oryginalnym grafie. Rozwiązanie oparte o ten pomysł jest więc zbyt wolne.

Wystarczy jednak zauważyć, że wierzchołki pośrednie dla różnych krawędzi można uwspólnić. Dokładniej: dla każdego wierzchołka u można stworzyć jego trzy kopie: u_2, u_1 oraz u_0. Będą one określały kolejno:

- u_2: wierzchołek pośredni oznaczający, że do wierzchołka u pozostały dwie krawędzie,
- u_1: wierzchołek pośredni oznaczający, że do wierzchołka u pozostała jedna krawędź,
- u_0: oryginalny wierzchołek u z grafu.

Taka definicja pozwala uwspólniać wierzchołki pośrednie i działa nawet wtedy, gdy wagi krawędzi są różne (rys. 3).



Rys. 3: Uwspólnienie wierzchołków pomiędzy końcami krawędzi o różnej długości.

Ostatecznie mamy gwarancję, że liczba wierzchołków w nowym grafie, w którym krawędzie są już jednostkowej długości, nie przekracza $3n$. Tak oto uzyskujemy rozwiązanie o złożoności $O(n^3 \log k)$, które na zawodach otrzymywało maksymalną punktację.

Aby uzyskać liczbę ścieżek długości co najwyżej ℓ, postępujemy analogicznie jak w rozwiązaniu dla jednostkowych wag krawędzi, z tą tylko różnicą, że do wierzchołka startowego podłączone są jedynie oryginalne wierzchołki grafu (z indeksem 0) oraz zliczamy ścieżki kończące się tylko w tych wierzchołkach.

Implementację takiego rozwiązania można odnaleźć w plikach `wyc.cpp` oraz `wyc0.pas`.

XXVII Międzynarodowa Olimpiada Informatyczna,

Ałmaty, Kazachstan 2015