Tomasz Waleń **Jakub Pawlewicz**
Treść zadania, Opracowanie Program

OI, Etap II, dzień drugi, 16.02.2006

Listonosz

Listonosz Bajtazar codziennie musi odwiedzić wszystkie ulice swojego rejonu i dostarczyć listy. Wszystkie ulice są jednokierunkowe i łączą (parami różne) skrzyżowania. Parę skrzyżowań mogą łączyć co najwyżej dwie ulice: jedna w jednym, a druga w drugim kierunku. Skrzyżowania są ponumerowane od 1 do n.

Bajtazar rozpoczyna i kończy trasę w centrali poczty Bajtockiej, przy skrzyżowaniu nr 1. Od dawien dawna Bajtazar sam wybierał trasę, którą obchodził swój rejon, jednak ostatnio dyrekcja poczty wydała nowe rozporządzenie, ograniczające swobodę wyboru tras. Każdemu listonoszowi przydzielono pewien zestaw fragmentów trasy — zbiór sekwencji skrzyżowań. Bajtazar musi wybrać taką trasę, która:

- *prowadzi każdą ulicą dokładnie raz,*
- *zawiera w sobie każdą z zadanych sekwencji (jako spójny podciąg),*
- *rozpoczyna się i kończy na skrzyżowaniu nr 1,*

Niestety, możliwe jest, że dyrekcja wydała rozporządzenie, dla którego nie istnieje trasa Bajtazara spełniająca wymogi, np. wymaga ono w jednej ze swoich sekwencji pójścia drogą, która nie istnieje. Pomóż Bajtazarowi i napisz program, który sprawdzi czy poprawna trasa istnieje, a jeśli tak, to wyznaczy ją.

Zadanie

Napisz program który:

- *wczyta ze standardowego wejścia opis ulic i przydzielone sekwencje,*
- *sprawdzi czy Bajtazar może obejść swój rejon, tak żeby odwiedzić każdą ulicę dokładnie raz oraz spełnić wszystkie zalecenia dyrekcji,*
- *wypisze na standardowe wyjście znalezioną trasę lub stwierdzi, że taka trasa nie istnieje.*

Wejście

W pierwszym wierszu standardowego wejścia zapisanie są dwie liczby całkowite n i m oddzielone pojedynczym odstępem, $2 \leqslant n \leqslant 50\,000$, $1 \leqslant m \leqslant 200\,000$, odpowiednio liczba skrzyżowań i ulic. W kolejnych m wierszach znajdują się opisy ulic: dwie liczby całkowite a, b, oddzielone pojedynczym odstępem, $1 \leqslant a,b \leqslant n$, $a \neq b$, oznaczają, że ze skrzyżowania a do b prowadzi (jednokierunkowa) ulica. Każda (uporządkowana) para a,b pojawia się w danych co najwyżej raz. W kolejnym wierszu zapisana jest liczba t, $0 \leqslant t \leqslant 10\,000$, oznaczająca liczbę nakazanych sekwencji. W kolejnych t wierszach zapisane są opisy sekwencji. Opis sekwencji składa się z liczby k, $2 \leqslant k \leqslant 200\,000$, oraz ciągu $v_1,\ldots,v_k$ numerów skrzyżowań. Liczby w wierszu są pooddzielane pojedynczymi odstępami. Sumaryczna długość wszystkich sekwencji nie przekracza $1\,000\,000$.

Wyjście

Twój program powinien wypisać w pierwszym wierszu wyjścia:

- `TAK` — *jeśli istnieje trasa spełniająca warunki zadania,*
- `NIE` — *jeśli taka trasa nie istnieje.*

W przypadku odpowiedzi `TAK`*, w kolejnych wierszach należy zapisać opis znalezionej trasy. Jeśli jest wiele takich tras, można wypisać dowolną z nich. Opis powinien składać się z sekwencji skrzyżowań kolejno odwiedzanych na trasie listonosza — $m+1$ liczb: $v_1, \ldots, v_{m+1}$, każdej wypisanej w osobnym wierszu, takich że:*

- $v_1 = v_{m+1} = 1$,
- v_i *i* v_{i+1} *(dla* $1 \leqslant i \leqslant m$*) są połączone ulicami,*
- *każda ulica występuje na liście dokładnie raz,*
- *trasa zawiera, jako spójne podciągi, wszystkie nakazane przez dyrekcję sekwencje.*

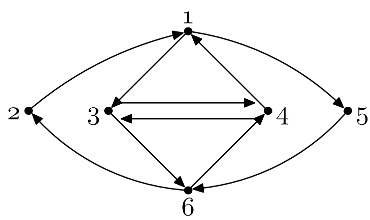
Przykład

Dla danych wejściowych:

```
6 10
1 5
1 3
4 1
6 4
3 6
3 4
4 3
5 6
6 2
2 1
4
3 1 5 6
3 3 4 3
4 4 3 6 4
3 5 6 2
```

poprawnym wynikiem jest:

```
TAK
1
3
4
3
6
4
1
5
6
2
1
```



Rozwiązanie

Przedstawmy sytuację opisaną w zadaniu jako graf, w którym wierzchołkami są skrzyżowania, a krawędziami — ulice. Zadanie polega na wyznaczeniu cyklu przechodzącego przez każdą krawędź grafu dokładnie raz, który spełnia zadane dodatkowe warunki: przez wybrane sekwencje krawędzi musimy przejść w określonym porządku.

Jeśli zbiór sekwencji byłby pusty, zadanie wymagałoby jedynie wyznaczenia cyklu Eulera, co potrafimy zrobić szybko i prosto (patrz na przykład [27] bądź [18]). W dalszej części opisu pokażemy jak zmodyfikować graf, by problem sprowadzić do znajdowania zwykłego cyklu Eulera.

Rozwiązanie dla jednej sekwencji

Rozpocznijmy od przypadku, gdy mamy tylko jedną sekwencję, czyli $t = 1$. Wówczas możemy zmodyfikować graf, usuwając z niego wszystkie krawędzie sekwencji i wprowadzając w zamian jedną krawędź — łączącą początek pierwszej krawędzi sekwencji z końcem ostatniej. W ten sposób każdy cykl Eulera w zmodyfikowanym grafie będzie odpowiadał cyklowi Eulera w grafie oryginalnym zawierającemu sekwencję. Jedynym (być może) skomplikowanym punktem powyższej procedury jest konieczność zapamiętania, że nowa krawędź zastępuje całą sekwencję krawędzi z oryginalnego grafu. Ta informacja będzie nam potrzebna przy wypisywaniu wyniku.

Schemat takiego rozwiązania możemy zapisać algorytmicznie:

- niech $G = (V, E)$ oznacza graf skrzyżowań, a jedyna zadana sekwencja ma postać $p_1, \ldots, p_k$.
- usuń z E krawędzie z sekwencji: $\{(p_i, p_{i+1}) : 1 \leqslant i < k\}$,
- dodaj do E krawędź (p_1, p_k) z etykietą $p_1, \ldots, p_k$,
- oblicz cykl Eulera w zmodyfikowanym grafie.

Ten sam sposób można zastosować w przypadku, gdy mamy wiele sekwencji, które są krawędziowo rozłączne. Każdą z nich zastępujemy krawędzią i wyznaczamy w zmienionym grafie cykl Eulera.

Ponieważ cykl Eulera można obliczyć w czasie $O(|V| + |E|)$, więc całe rozwiązanie wymaga czasu liniowego względem rozmiaru danych wejściowych.

Rozwiązanie dla ogólnego przypadku

Pozostaje nam do rozważenia przypadek, gdy zadane sekwencje mają wspólne krawędzie. W pierwszej kolejności sprawdzimy, czy sekwencje nie są ze sobą sprzeczne (zauważmy na przykład, że nie jest możliwe jednoczesne spełnienie sekwencji $(1, 2, 3)$ i $(1, 2, 4)$). Użyjemy w tym celu dwóch tablic *nast* i *poprz* indeksowanych krawędziami grafu. W $nast[e]$ zapiszemy informację, jaką krawędź musimy przejść bezpośrednio po krawędzi e, jeśli takie ograniczenie wynika z zadanych sekwencji. Analogicznie wypełnimy tablicę *poprz*.

for $e \in E$ **do** $nast[e] := \textbf{nil}, poprz[e] := \textbf{nil}$;
for $s := 1$ **to** t **do** **begin**

```
  { Niech p_1,...,p_k oznacza s–tą sekwencję }
  for i := 1 to k−2 do
    e_1 := (p_i, p_{i+1}); e_2 := (p_{i+1}, p_{i+2});
    if e_1 ∉ E or e_2 ∉ E then
      return "BŁĄD - krawędzi nie ma w grafie";
    if nast[e_1] = nil then nast[e_1] := e_2
    else if nast[e_1] ≠ e_2 then return "BŁĄD - sprzeczne sekwencje";
    if poprz[e_2] = nil then poprz[e_2] := e_1
    else if poprz[e_2] ≠ e_1 then return "BŁĄD - sprzeczne sekwencje";
  end;
end;
return "OK";
```

Jeśli powyższa procedura zwróci błąd, to nie istnieje cykl zawierający wszystkie zadane sekwencje. Jeśli procedura zakończy się sukcesem, to możemy z tablic *nast*/*poprz* przygotować nowy zestaw sekwencji — równoważny oryginalnemu, ale już krawędziowo rozłączny. Wystarczy „odczytać” ścieżki zapisane na przykład w tablicy *nast* — są to zsumowane zadane sekwencje.

Trudności techniczne

Jakkolwiek rozwiązanie zadania nie jest zbyt trudne do wymyślenia, jednak poprawna jego implementacja może sprawiać pewne trudności.

Pierwszym problemem do rozwiązania jest przygotowanie tablic indeksowanych krawędziami grafu. Można do tego celu wykorzystać strukturę `map` z biblioteki STL, jednak w takim przypadku dostęp do jednej pozycji tablicy wymaga czasu $O(\log n)$. Innym rozwiązaniem jest „ponumerowanie” krawędzi przez przypisanie im liczb z zakresu $1,\ldots,|E|$. Taką numerację możemy uzyskać, sortując krawędzie jako pary wierzchołków — ze względu na ograniczony zbiór wierzchołków możemy wykonać to w czasie liniowym. Jeśli każdą krawędź sekwencji uzupełnimy o jej numer, późniejsze odwołania do tablic *nast*/*poprz* można wykonać w czasie $O(1)$.

Innym problemem są przypadki szczególne. Aby poprawnie rozwiązać zadanie, trzeba uniknąć wielu pułapek:

- Graf spójny po modyfikacji może składać się z więcej niż jednej spójnej składowej. Jeśli co najmniej dwie z nich zawierają krawędzie, to rozwiązanie nie istnieje.
- Problemem są także cykle powstałe przez zsumowanie sekwencji. Jeśli krawędzie z tablic *nast*/*poprz* tworzą cykl (na przykład, jak dla sekwencji $S_1 = (1,2,3)$ i $S_2 = (2,3,1,2)$), to:
  - jeśli ten cykl nie zawiera wszystkich krawędzi, to rozwiązanie na pewno nie istnieje;
  - jeśli cykl zawiera wszystkie krawędzie grafu, to trzeba jeszcze sprawdzić, czy ograniczenia nałożone przez sekwencje pozwalają listonoszowi zacząć i zakończyć trasę w centrali poczty.

Kolejną trudnością jest konieczność zapisu „długich” etykiet krawędzi stanowiących skróty ścieżek oryginalnego grafu oraz ich odczytywanie podczas wypisywania odpowiedzi.

Testy

Zadanie testowane było na następującym zestawie danych testowych.

Nazwa	n	m	t	∑k	Opis
lis1a.in	3	3	2	4	na sprawdzanie warunku Eulera (odp. NIE)
lis1b.in	4	6	2	4	prosty test poprawnościowy
lis2a.in	5	7	2	6	sprzeczne sekwencje (odp. NIE)
lis2b.in	11	20	3	10	na sklejanie ścieżek
lis3a.in	7	9	2	6	sprzeczne sekwencje (odp. NIE)
lis3b.in	7	14	4	12	koniec ścieżki w początku ścieżki
lis4a.in	5	6	2	8	ścieżki po sklejeniu dają cykl (odp. NIE)
lis4b.in	7	12	10	24	na sklejanie ścieżek
lis5a.in	7	9	2	6	rozspójnienie grafu po skróceniu sekwencji (odp. NIE)
lis5b.in	20	100	50	200	test losowy
lis6.in	100	500	50	200	test losowy
lis7.in	500	2 000	1 000	10 000	test losowy
lis8.in	2 000	10 000	10 000	50 000	test losowy
lis9.in	10 000	50 000	9 999	200 000	test losowy
lis10.in	49 999	60000	100	400	test losowy
lis11.in	450	199 998	500	4 000	test losowy
lis12.in	300	30 000	10 000	100 000	test losowy
lis13.in	40 000	200 000	5	1 000 000	test losowy
lis14.in	50 000	200 000	10 000	1 000 000	test losowy