Marek Cygan, Wojciech Rytter
Treść zadania, Opracowanie

Krzysztof Leszczyński
Program

Dostępna pamięć: 256 MB. *OI, etap III, dzień drugi, 29.03.2012*

Bezpieczeństwo minimalistyczne

Dana jest mapa sieci ulic Mafiogrodu. Sieć ta składa się ze skrzyżowań i łączących je dwukierunkowych ulic. Ulice nie przecinają się nigdzie poza skrzyżowaniami, choć mogą prowadzić tunelami lub estakadami. Między każdą parą skrzyżowań przebiega co najwyżej jedna ulica. Na każdym skrzyżowaniu v *jest posterunek policji, w którym jest* $p(v)$ *policjantów. Aby ulica łącząca skrzyżowania* u *i* v *mogła zostać uznana za bezpieczną, w sumie z obu stron tej ulicy musi być co najmniej* $b(u,v)$ *policjantów. Początkowo dla każdej ulicy zachodzi* $p(u) + p(v) \geqslant b(u,v)$.

Z powodu kryzysu burmistrz Bajtazar wydał zarządzenie o **bezpieczeństwie minimalistycznym** *mówiące, że:*

- *z każdego posterunku należy zwolnić pewną liczbę (być może zero) policjantów (liczbę policjantów zwolnionych z posterunku przy skrzyżowaniu* v *oznaczmy przez* $z(v)$*),*
- *po przeprowadzeniu zwolnień, łączna liczba policjantów w posterunkach na końcach każdej ulicy, łączącej skrzyżowania* u *i* v*, ma być równa dokładnie* $b(u,v)$*, czyli:*

$$p(u) - z(u) + p(v) - z(v) = b(u,v).$$

Z powyższych zasad nie wynika jednoznacznie, ilu policjantów ma zostać zwolnionych. Bajtazar zastanawia się, jaka jest najmniejsza i największa możliwa liczba zwolnionych policjantów (suma wartości z *dla wszystkich skrzyżowań) spełniająca powyższe zasady.*

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite n *oraz* m $(1 \leqslant n \leqslant 500\,000,\ 0 \leqslant m \leqslant 3\,000\,000)$*, oddzielone pojedynczym odstępem, oznaczające liczbę skrzyżowań oraz liczbę ulic w Mafiogrodzie. Skrzyżowania są ponumerowane od 1 do* n*. W drugim wierszu zapisanych jest dokładnie* n *nieujemnych liczb całkowitych pooddzielanych pojedynczymi odstępami. Są to liczby policjantów aktualnie zatrudnionych w kolejnych posterunkach, czyli wartości* $p(1), p(2), \ldots, p(n)$ $(0 \leqslant p(i) \leqslant 10^6)$.

Każdy z kolejnych m *wierszy zawiera opis jednej ulicy dwukierunkowej. Opis taki składa się z trzech liczb całkowitych* u_i, v_i, $b(u_i,v_i)$ $(1 \leqslant u_i, v_i \leqslant n,\ u_i \neq v_i,\ 0 \leqslant b(u_i,v_i) \leqslant 10^6)$ *oznaczających odpowiednio numery skrzyżowań będących końcami danej ulicy oraz minimalną liczbę policjantów, którzy muszą być zatrudnieni w sumie na obu końcach danej ulicy.*

W testach wartych 56% punktów zachodzą dodatkowe warunki $n \leqslant 2\,000$ *oraz* $m \leqslant 8\,000$.

Wyjście

Jeśli spełnienie wymogów zarządzenia Bajtazara jest możliwe, Twój program powinien wypisać na standardowe wyjście dokładnie jeden wiersz zawierający dwie liczby całkowite oddzielone *pojedynczym odstępem. Mają to być najmniejsza oraz największa liczba policjantów, jakich należy zwolnić, aby spełnić wymogi zarządzenia.*

Jeśli spełnienie wymogów zarządzenia nie jest możliwe, Twój program powinien wypisać jeden wiersz zawierający słowo `NIE`*.*

Przykład

Dla danych wejściowych:

```
3 2
5 10 5
1 2 5
2 3 3
```

poprawnym wynikiem jest:

```
12 15
```

natomiast dla danych:

```
3 3
1 1 1
1 2 1
1 3 1
3 2 1
```

poprawnym wynikiem jest:

```
NIE
```

Rozwiązanie

W sformułowaniu zadania w naturalny sposób występuje nieskierowany graf, którego wierzchołki odpowiadają skrzyżowaniom, natomiast krawędzie — ulicom Mafiogrodu. Zauważmy, że każdą spójną składową grafu możemy rozważać niezależnie, dlatego też bez straty ogólności załóżmy, że graf dany w zadaniu jest spójny.

Rozważmy pierwszy test przykładowy z treści zadania i przez f_1, f_2, f_3 oznaczmy liczbę policjantów zatrudnionych na poszczególnych posterunkach już po przeprowadzonej redukcji etatów. Krawędzie występujące w grafie wyznaczają równania:

$$f_1 + f_2 = 5,$$
$$f_2 + f_3 = 3.$$

Jako że Bajtazar zakazał zatrudniania dodatkowych policjantów, a liczba policjantów nie może być ujemna, dodatkowo mamy ograniczenia:

$$0 \leqslant f_1 \leqslant p(1) = 5,$$
$$0 \leqslant f_2 \leqslant p(2) = 10,$$
$$0 \leqslant f_3 \leqslant p(3) = 5.$$

Celem zadania jest wyznaczenie całkowitych wartości f_1, f_2, f_3, w taki sposób, aby spełnione były powyższe równania oraz nierówności, a liczba $f_1 + f_2 + f_3$, czyli ostateczna liczba zatrudnionych policjantów, była możliwie duża (lub mała). Oczywiście powyższe rozumowanie możemy zastosować dla grafu dowolnej wielkości. Wówczas krawędź uv wyznacza *równanie krawędziowe* $f_u + f_v = b(u,v)$, a każdy wierzchołek musi spełniać *nierówność wierzchołkową* $0 \leqslant f_v \leqslant p(v)$.

Rozwiązanie wolne

Zauważmy, że jeśli ustalimy docelową liczbę policjantów pracujących na pierwszym posterunku, to równania krawędziowe jednoznacznie wyznaczą nam liczby policjantów zatrudnionych na pozostałych posterunkach, co wynika z przyjętego założenia o spójności grafu. Żądane liczby policjantów można w praktyce wyznaczyć za pomocą dowolnego algorytmu przeszukiwania grafu, np. metody BFS czy DFS. Prowadzi nas to do rozwiązania wolnego, zaimplementowanego w pliku `bezs0.cpp`, o złożoności $O(p(1)(n+m))$, gdyż docelowa liczba policjantów na pierwszym posterunku musi być całkowita i należeć do przedziału $[0, p(1)]$.

Możemy również wykorzystać sytuację, w której początkowa liczba policjantów na poszczególnych posterunkach znacznie się różni. Zamiast wyznaczać docelową liczbę policjantów na pierwszym posterunku, możemy użyć posterunku o najmniejszej liczbie początkowo zatrudnionych policjantów. Takie rozwiązanie znajdziemy w pliku `bezs1.cpp`.

Omawiane rozwiązania pozwalały na uzyskanie 56 punktów.

Rozwiązanie wzorcowe

Zamiast rozpoczynać nasze rozwiązanie od wyznaczenia wartości f_1, potraktujmy tę wartość jako zmienną, czyli przyjmijmy $f_1 = x$. Rozpatrując dowolne drzewo rozpinające grafu ukorzenione w wierzchołku 1, za pomocą równań krawędziowych możemy jednoznacznie wyznaczyć wartości f_i jako liniowe funkcje zmiennej x. Jeśli dla wierzchołka v określiliśmy już funkcję $f_v(x) = a_v x + c_v$, a wierzchołek u jest synem wierzchołka v w drzewie, to

$$f_u(x) = -a_v x + (b(u, v) - c_v).$$

Co więcej, zauważmy, że każda z tych funkcji będzie miała postać $f_i(x) = x + c_i$ lub $f_i(x) = -x + c_i$.

Graf dany na wejściu nie musi być jednak drzewem, co oznacza, że musimy upewnić się, że równania krawędziowe dla krawędzi spoza drzewa rozpinającego również są spełnione. Rozważmy krawędź uv oraz załóżmy, że funkcje dla wierzchołków u oraz v mają postać $f_u(x) = a_u x + c_u$ oraz $f_v(x) = a_v x + c_v$. Mamy dwa następujące przypadki:

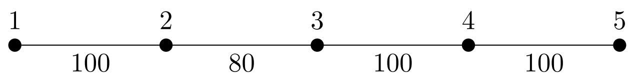
- jeśli $a_u \neq a_v$, to równanie $f_u(x) + f_v(x) = b(u, v)$ przyjmuje postać $c_u + c_v = b(u, v)$. W tej sytuacji w równaniu nie występuje zmienna x, co oznacza, że równanie to jest zawsze spełnione lub też zawsze sprzeczne (w tym drugim wypadku wiemy, że nie istnieje rozwiązanie zadania);
- jeśli natomiast $a_u = a_v$, to równanie krawędziowe przyjmuje postać

  $$x = \frac{b(u, v) - c_u - c_v}{2a_u},$$

  co pozwala nam jednoznacznie wyznaczyć wartość x i sprawdzić, czy spełnione są dla niej pozostałe równania krawędziowe, jak również nierówności wierzchołkowe. W tym przypadku zadanie ma co najwyżej jedno rozwiązanie, a w związku z tym najmniejsza i największa liczba zwolnionych policjantów będą miały dokładnie tę samą wartość.

Pozostały nam do rozważenia nierówności wierzchołkowe. Każdy wierzchołek nakłada na zmienną x dwie nierówności, mianowicie dla wierzchołka v oraz funkcji $f_v(x) = a_v x + c_v$ mamy $0 \leqslant a_v x + c_v \leqslant p(v)$. Każdy wierzchołek wyznacza zatem pewien przedział dopuszczalnych wartości dla zmiennej x. W przypadku $a_v = 1$ mamy przedział $[-c_v, p(v) - c_v]$, natomiast w przypadku $a_v = -1$ — przedział $[c_v - p(v), c_v]$. Przecięcie wszystkich przedziałów pochodzących z nierówności wierzchołkowych pozwala nam wyznaczyć przedział $[A, B]$ reprezentujący dopuszczalne wartości dla zmiennej x. Oznacza to, że jeśli zmiennej x przypiszemy dowolną całkowitą wartość z przedziału $[A, B]$, to automatycznie spełnione będą zarówno wszystkie równania krawędziowe, jak i nierówności wierzchołkowe.

Przykładowo, w przypadku poniższego grafu (pojedyncza ścieżka; na krawędziach podano wartości $b(i, j)$) funkcje f_i oraz dopuszczalne przedziały mają następującą postać:



i	1	2	3	4	5
$p(i)$	90	50	100	40	90
$f_i(x)$	x	$100 - x$	$x - 20$	$120 - x$	$x - 20$
przedział x	$[0, 90]$	$[50, 100]$	$[20, 120]$	$[80, 120]$	$[20, 110]$

Przecięcie wszystkich przedziałów to $[80, 90]$. Jeśli do grafu dołożymy krawędź pomiędzy wierzchołkami 2 oraz 4 z wartością $b(2, 4) = 50$, to będziemy mieli tylko jedno rozwiązanie, dla $x = 85$. Gdybyśmy natomiast, zamiast tej krawędzi, dodali krawędź 2–5 z wartością $b(2, 5) = 100$, to nie mielibyśmy już żadnego rozwiązania.

Nie powiedzieliśmy jeszcze, w jaki sposób możemy wyznaczyć najmniejszą i największą liczbę policjantów, których należy zwolnić. W tym celu wystarczy zauważyć, że całkowita liczba zwolnionych policjantów wyraża się wzorem $\sum_{i=1}^{n}(p(i) - f_i(x))$. Jako że suma funkcji liniowych jest funkcją liniową, najmniejsza i największa jej wartość będzie osiągana na krańcach przedziału $[A, B]$, dzięki czemu wystarczy obliczyć liczbę zwolnionych policjantów jedynie dla wartości $x = A$ oraz $x = B$.

Przedstawione rozwiązanie ma złożoność $O(n + m)$. Jego implementację można znaleźć w pliku `bez.cpp`.

Testy

W poniższej tabeli s oznacza liczbę spójnych składowych, natomiast p to najmniejsza z liczb policjantów na poszczególnych posterunkach (tj. minimum z liczb $p(i)$).

Nazwa	**n**	**m**	**s**	**p**
bez1a.in	10	12	2	5
bez1b.in	10	9	1	10
bez1c.in	10	17	1	2
bez1d.in	1	0	1	10
bez2a.in	200	465	2	114
bez2b.in	20	19	1	1 000
bez2c.in	20	49	1	193
bez2d.in	200	411	46	160
bez3a.in	2 000	7 897	15	33
bez3b.in	20	43	1	60
bez3c.in	20	54	1	106
bez3d.in	2 000	5 019	410	14
bez4a.in	2 000	7 920	7	11 959
bez4b.in	20	19	1	1 000
bez4c.in	20	53	1	204
bez4d.in	2 000	5 000	408	30
bez5a.in	500 000	1 215 391	1 881	606
bez5b.in	20	19	1	1 000
bez5c.in	500 000	1 299 594	101 128	2
bez6a.in	100 000	249 888	198	136
bez6b.in	20	45	1	184
bez6c.in	100 000	259 677	20 224	184
bez7a.in	500 000	1 999 856	1 845	1 472
bez7b.in	20	56	1	97
bez7c.in	500 000	499 999	1	1 000 000