Jakub Radoszewski	Marek Cygan, Łukasz Kowalik	Paweł Parys
Treść zadania	Opracowanie	Program

OI, Etap II, dzień próbny, 14.02.2006

Szkoły

W Bajtocji znajduje się n szkół, z których każda ma przypisany numer m, $1 \leqslant m \leqslant n$. Poprzedni król Bajtocji w ogóle nie zwracał uwagi na porządek w numeracji szkół, pozwalając każdej nowowybudowanej szkole wybrać dla siebie dowolny numer z przedziału od 1 do n. Przez to w kraju mogło powstać wiele szkół o tym samym numerze, a niektóre numery z przedziału od 1 do n mogły nie zostać wykorzystane.

Nowy król Bajtocji postanowił przywrócić porządek i dokonać takiego przenumerowania szkół, żeby teraz każdy numer był wykorzystany dokładnie raz. Niestety nie jest to proste zadanie, gdyż większość szkół niechętnie poddałaby się zmianie numeru.

Król wysłał swoich informatorów do poszczególnych szkół, aby dowiedzieli się, na jak dużą zmianę numeru poszczególne szkoły mogą się zgodzić. Co więcej, każda szkoła określiła koszt c dokonania zmiany swojego numeru o 1. Stąd całkowity koszt dokonania zmiany numeru danej szkoły wynosi $c \cdot |m - m'|$, gdzie m oznacza stary, a m' nowy numer danej szkoły. Oczywiście liczba m' musi się mieścić w podanym wcześniej przedziale tolerancji dla danej szkoły.

Król — dostawszy wyżej opisane informacje — chciałby się dowiedzieć, czy jest możliwy do przywrócenia porządek w numeracji szkół (zakładając przestrzeganie przedziałów tolerancji wszystkich szkół), a jeżeli tak, to jaki jest minimalny koszt takiego przenumerowania. Dlatego poprosił Ciebie — nadwornego informatyka — o podanie mu tej informacji na podstawie danych o szkołach, które Ci dostarczył.

Zadanie

Napisz program, który:

- *wczyta ze standardowego wejścia aktualne numery szkół w Bajtocji, ich przedziały tolerancji dotyczące zmiany numeru oraz koszty zmiany numerów poszczególnych szkół o 1,*
- *sprawdzi, czy jest możliwe przenumerowanie szkół spełniające wszystkie wymienione wcześniej warunki i jeżeli tak — obliczy minimalny koszt takiej zmiany,*
- *wypisze wynik na standardowe wyjście.*

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą n ($1 \leqslant n \leqslant 200$), oznaczającą liczbę szkół w Bajtocji. W kolejnych n wierszach znajdują się opisy poszczególnych szkół. Wiersz o numerze $i+1$ ($1 \leqslant i \leqslant n$) zawiera cztery liczby całkowite m_i, a_i, b_i oraz k_i ($1 \leqslant a_i \leqslant m_i \leqslant b_i \leqslant n$, $1 \leqslant k_i \leqslant 1\ 000$), oddzielone pojedynczymi odstępami. Liczby te oznaczające odpowiednio: aktualny numer i-tej szkoły, początek i koniec przedziału tolerancji i-tej szkoły odnośnie zmiany numeru (jest to przedział domknięty, czyli nowy numer i-tej szkoły m'_i musi spełniać nierówność $a_i \leqslant m'_i \leqslant b_i$) oraz koszt zamiany numeru i-tej szkoły o 1.

Wyjście

Jeżeli przenumerowanie szkół spełniające podane wyżej warunki jest możliwe, program powinien wypisać jedną liczbę całkowitą k oznaczającą najmniejszy możliwy koszt zmiany numeracji. W przeciwnym przypadku na wyjściu powinno zostać wypisane słowo `NIE`.

Przykład

Dla danych wejściowych:

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

poprawnym wynikiem jest:

```
9
```

Rozwiązanie

Rozwiązanie wzorcowe

Skojarzenia

Sformułujmy zadanie w terminach teorii grafów. *Grafem dwudzielnym* nazywamy graf, w którym wierzchołki można podzielić na dwie rozłączne grupy, takie że krawędzie przebiegają jedynie pomiędzy wierzchołkami z różnych grup. Każdemu egzemplarzowi naszego problemu odpowiada następujący graf dwudzielny $G = (V, E)$:

- $V = S \cup N$, gdzie $S = \{s_1, \ldots, s_n\}$ oraz $N = \{1, \ldots, n\}$ oznaczają odpowiednio szkoły oraz numery, które zamierzamy przypisać szkołom.
- Krawędź między szkołą s_i oraz numerem k istnieje, jeżeli dana szkoła może zaakceptować ten numer, czyli zachodzi $a_i \leqslant k \leqslant b_i$. Wagą każdej krawędzi jest koszt odpowiedniego przypisania.

Skojarzeniem w grafie G nazywamy podzbiór $M \subseteq E$ taki, że żadne dwie krawędzie z M nie mają wspólnego wierzchołka. Dla każdego podzbioru $F \subseteq E$ określimy *wagę* F, jako sumę wag krawędzi należących do F — będziemy ją oznaczać $waga(F)$. Zadanie sprowadza się zatem do znalezienia minimalnej wagi skojarzenia o rozmiarze n.

Niech M będzie skojarzeniem w grafie G. Definiujemy graf skierowany $D(G,M)$ następująco:

- Wierzchołki i krawędzie w grafie $D(G,M)$ są takie same jak w grafie G, ale wszystkie krawędzie grafu $D(G,M)$ są skierowane.

- Jeżeli w grafie G istnieje krawędź o wadze w łącząca wierzchołki $v_1 \in S$ i $v_2 \in N$ oraz krawędź ta należy do skojarzenia M, wówczas w grafie $D(G,M)$ jest ona zorientowana z wierzchołka v_2 do v_1 i ma wagę $-w$ (ujemną!).

- Jeżeli w grafie G istnieje krawędź o wadze w łącząca wierzchołki $v_1 \in S$ i $v_2 \in N$ oraz krawędź ta nie należy do skojarzenia M, wówczas w grafie $D(G,M)$ jest ona zorientowana z wierzchołka v_1 do v_2 i ma wagę w.

Ścieżką rozszerzającą w grafie $D(G,M)$ nazywamy ścieżkę rozpoczynającą się w wierzchołku $u \in S$ i kończącą się w wierzchołku $v \in N$ taką, że żaden z wierzchołków u, v nie jest pokrywany przez skojarzenie M.

Opis algorytmu

Przedstawimy teraz algorytm służący do znajdowania skojarzenia maksymalnego rozmiaru i posiadającego minimalną wagę.

Zdefiniujemy operator różnicy symetrycznej dla zbiorów:

$$A \oplus B = (A \cup B) - (A \cap B) = (A - B) \cup (B - A).$$

Niech M_0 oznacza puste skojarzenie. Przez długość ścieżki będziemy rozumieć sumę wag jej krawędzi.

Algorytm wykonuje co najwyżej n kroków. W k-tym kroku obliczamy skojarzenie M_k (w skład którego wchodzi k krawędzi) w następujący sposób. Najpierw znajdujemy najkrótszą ścieżkę rozszerzającą P_k w grafie $D(G,M_{k-1})$ (powrócimy jeszcze do tego, jak to zrobić). Wyznaczamy skojarzenie $M_k = M_{k-1} \oplus P_k$, czyli zaliczamy do M_k krawędzie, które należą do M_{k-1} lub do P_k, ale nie do obu równocześnie. Jeżeli w którymś momencie okaże się, że ścieżka rozszerzająca nie istnieje, oznacza to, że szukane skojarzenie nie istnieje.

Należy jeszcze zwrócić uwagę na jedną rzecz: w grafie z wagami, w którym niektóre wagi są ujemne, mogą istnieć dowolnie krótkie ścieżki. Zdarza się to, gdy w grafie tym istnieje cykl o ujemnej wadze. Wtedy pojęcie „najkrótsza ścieżka” nie jest dobrze zdefiniowane. Jednak w naszym zadaniu ścieżka rozszerzająca nie może przechodzić 2 razy przez ten sam wierzchołek, ponieważ wierzchołek ten musiałby być incydentny z dwiema krawędziami skojarzenia, co jest niemożliwe.

Dlaczego to działa?

Teraz musimy pokazać, że powyższy algorytm zawsze znajduje skojarzenie o minimalnej wadze.

Przypuśćmy, że w grafie G istnieje skojarzenie rozmiaru $s \leqslant n$. Skorzystamy z indukcji matematycznej i pokażemy, że dla każdego $k = 1, \ldots, s$ algorytm znajdzie skojarzenie M_k zawierające k krawędzi oraz posiadające minimalną wagę spośród wszystkich skojarzeń rozmiaru k.

Rozważmy k-ty krok algorytmu. Z założenia indukcyjnego przyjmujemy, że skojarzenie M_{k-1} znalezione w poprzednim kroku algorytmu posiada $k-1$ krawędzi i ma ono minimalną wagę wśród skojarzeń o tej mocy.

Niech M'_k będzie skojarzeniem w G rozmiaru k o minimalnej wadze. Jak wygląda zbiór krawędzi $M_{k-1} \oplus M'_k$? Składa się on wyłącznie z cykli i ścieżek, gdyż stopień każdego wierzchołka jest w nim niewiększy niż 2. Dodatkowo, w każdej ze ścieżek liczby krawędzi ze zbioru M'_k i M_{k-1} mogą się różnić co najwyżej o 1 (ponieważ występują one na przemian). Ponieważ zbiór M'_k posiada o jedną krawędź więcej niż M_{k-1}, zatem musi istnieć co najmniej jedna ścieżka, w której liczba krawędzi ze zbioru M'_k jest o 1 większa niż liczba krawędzi z M_{k-1}. Oznaczmy taką ścieżkę przez P. Zauważmy, że $M'_k \oplus P$ oraz $M_{k-1} \oplus P$ są skojarzeniami, posiadającymi odpowiednio $k-1$ i k krawędzi.

Lemat 1 *Krawędzie z M'_k nienależące do P mają taką samą sumę wag, jak krawędzie z M_{k-1} nienależące do P. Inaczej mówiąc:*

$$waga(M'_k \setminus P) = waga(M_{k-1} \setminus P)$$

Dowód Załóżmy, że powyższa teza nie zachodzi. Wtedy musi zachodzić dokładnie jeden z warunków:

- $waga(M'_k \setminus P) > waga(M_{k-1} \setminus P)$, zatem

  $waga(M'_k) - waga(M'_k \cap P) > waga(M_{k-1} \setminus P))$, korzystając z rozłączności zbiorów

  $waga(M'_k) > waga((M_{k-1} \setminus P) \cup (M'_k \cap P))$ i dalej

  $waga(M'_k) > waga(M_{k-1} \oplus P)$, co oznacza sprzeczność, gdyż M'_k jest skojarzeniem liczności k o minimalnej wadze

- rozważając analogicznie drugi przypadek otrzymujemy

  $waga(M'_k \setminus P) < waga(M_{k-1} \setminus P)$, zatem

  $waga(M'_k \setminus P) < waga(M_{k-1}) - waga(M_{k-1} \cap P)$, co oznacza

  $waga((M'_k \setminus P) \cup (M_{k-1} \cap P)) < waga(M_{k-1})$ i dalej

  $waga(M'_k \oplus P) < waga(M_{k-1})$, co doprowadza do sprzeczności, gdyż M_{k-1} jest skojarzeniem liczności $k-1$ o minimalnej wadze

■

Teraz zauważmy, że P jest ścieżką rozszerzającą w grafie $D(G, M_{k-1})$, a z powyższego lematu wynika, że dla wag krawędzi z grafu $D(G, M_{k-1})$ zachodzi równość: $waga(M'_k) = waga(M_{k-1}) + waga(P)$ (jest to odpowiednio przekształcona postać równości z lematu). W k-tym kroku algorytmu znajdujemy pewną ścieżkę rozszerzającą — oznaczmy ją P_{alg}, i przyjmujemy $M_k = M_{k-1} \oplus P_{alg}$. Ponieważ znaleziona ścieżka jest najkrótszą ścieżką rozszerzającą, zatem $waga(P_{alg}) \leqslant waga(P)$. Otrzymujemy więc, że $waga(M_k) = waga(M_{k-1}) + waga(P_{alg}) \leqslant waga(M_{k-1}) + waga(P) = waga(M'_k)$. Zatem M_k ma minimalną wagę spośród wszystkich skojarzeń rozmiaru k.

Implementacja

W zadaniu wystarczy zatem zaimplementować algorytm znajdowania najkrótszych ścieżek rozszerzających w grafach postaci $D(G, M_k)$. Ponieważ w tych grafach mogą znajdować się krawędzie z ujemnymi wagami, nie możemy w tym celu posłużyć się algorytmem Dijkstry. Zamiast tego użyjemy algorytmu Bellmana-Forda[1], który pozwala znaleźć najkrótszą ścieżkę

[1]Zobacz np. w: [18]

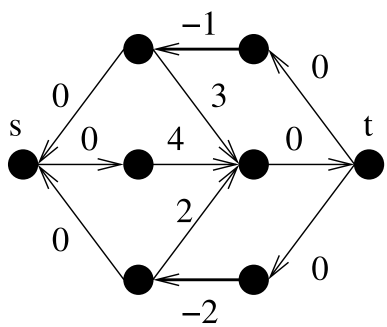
rozszerzającą w grafie z dowolnymi wagami, pod warunkiem, że nie występują w nim cykle o ujemnych wagach. Sprawdźmy, czy nasz graf spełnia ten warunek.

Przypuśćmy, że w grafie $D(G,M_k)$ występuje cykl C o ujemnej wadze. Wówczas dokładnie połowa krawędzi tego cyklu należy do skojarzenia M_k. Ponieważ po modyfikacjach wag w grafie $D(G,M_k)$ wszystkie krawędzie należące do skojarzenia mają wagi ujemne, a pozostałe krawędzie mają wagi dodatnie, zatem suma oryginalnych wag krawędzi należących do skojarzenia jest większa niż suma wag pozostałych krawędzi. Wtedy jednak $M_k \oplus C$ jest skojarzeniem rozmiaru k takim, że $waga(M_k \oplus C) < waga(M_k)$. Jednak to nie może zachodzić, ponieważ udowodniliśmy już, że M_k posiada minimalną wagę spośród wszystkich skojarzeń rozmiaru k.

Zatem możemy użyć algorytmu Bellmana-Forda. Jego złożoność czasowa wynosi $O(mn)$ dla grafu z m krawędziami oraz n wierzchołkami. Ponieważ liczba krawędzi może być co najwyżej rozmiaru $O(n^2)$, złożoność algorytmu Bellmana-Forda to $O(n^3)$, zatem złożoność czasowa całego algorytmu wynosi $O(n^4)$.

Aby znaleźć ścieżkę powiększającą w grafie $D(G,M)$ musimy dodać do grafu dwa dodatkowe wierzchołki. Jeden z nich będziemy nazywali źródłem, a drugi ujściem (terminologia pochodzi z zagadnień szukania maksymalnego przepływu). Źródło będziemy oznaczać jako wierzchołek s, a ujście jako wierzchołek t. Do grafu $D(G,M)$ dodajmy następujące krawędzie o wadze 0:

- pomiędzy wierzchołkiem s a każdym wierzchołkiem $v \in S$, jeśli wierzchołek v nie jest incydentny z żadną krawędzią należącą do skojarzenia, to krawędź jest skierowana od wierzchołka s do wierzchołka v, w przeciwnym wypadku krawędź jest skierowana w kierunku wierzchołka s,
- pomiędzy wierzchołkiem t, a każdym wierzchołkiem $v \in N$, jeśli wierzchołek v nie jest incydentny z żadną krawędzią należącą do skojarzenia, to krawędź jest skierowana od wierzchołka v do wierzchołka t, w przeciwnym wypadku krawędź jest skierowana w kierunku wierzchołka v.



Rys. 1: Przykładowy graf ze skojarzeniem o liczności 2

Powstały graf będziemy oznaczać $D'(G,M)$. Nietrudno zauważyć, że ścieżka powiększająca w grafie $D(G,M)$ odpowiada ścieżce w grafie $D'(G,M)$ prowadzącej od wierzchołka s do wierzchołka t. Po znalezieniu każdej nowej ścieżki powiększającej, odwracamy znajdujące się na niej krawędzie i zmieniamy ich wagi na przeciwne.

Inne poprawne rozwiązania

Wiadomo, że maksymalne skojarzenie w grafie może zostać znalezione za pomocą algorytmów znajdujących maksymalny przepływ w sieci. Również problem znajdowania skojarzenia o minimalnej wadze spośród wszystkich skojarzeń danego rozmiaru może zostać sprowadzony do znajdowania przepływu o danym rozmiarze i minimalnym koszcie. Aby rozwiązać zadanie, można więc zmodyfikować odpowiednio algorytm Forda-Fulkersona. W ten sposób otrzymamy algorytm o takiej samej złożoności czasowej ($O(n^4)$) jak algorytm wzorcowy.

Szybsze rozwiązania

Istnieją również szybsze algorytmy znajdowania skojarzenia o minimalnej wadze (jak również przepływu o minimalnym koszcie). Możemy zmodyfikować nasz algorytm, aby do znajdowania najkrótszych ścieżek powiększających używać algorytmu Dijkstry, zamiast algorytmu Bellmana-Forda. Dzięki temu otrzymamy algorytm o złożoności czasowej $O(n(m+n^2)) = O(n^3)$.

Wiadomo, że algorytm Dijkstry możemy stosować tylko i wyłącznie dla grafów o nieujemnych wagach krawędzi (wymyślenie przykładu grafu z ujemnymi wagami krawędzi, dla którego algorytm Dijkstry znajdzie niepoprawne rozwiązanie pozostawiamy Czytelnikowi jako ćwiczenie), a w grafie $D'(G,M)$ krawędzie należące do skojarzenia mogą mieć ujemne wagi. Aby pokonać tę trudność wprowadźmy pojęcie potencjału. Potencjał wierzchołka v będziemy oznaczać jako $\pi(v)$. Dla każdej krawędzi w grafie $D'(G,M)$ zdefiniujmy nową wagę $waga'(v_1,v_2) = waga(v_1,v_2) + \pi(v_1) - \pi(v_2)$. Zauważmy, że dla każdej ścieżki P z wierzchołka s do wierzchołka t zachodzi:

$$waga'(P) = waga(P) + \pi(s) - \pi(t),$$

czyli dla zadanego potencjału wagi te różnią się jedynie o stały składnik. Wynika z tego, że najkrótsza ścieżka powiększająca dla nowych wag jest również najkrótszą ścieżką powiększającą dla poprzednich kosztów.

Możemy sobie zadać pytanie, co daje nam wprowadzenie potencjału π. Otóż możemy tak dobrać jego wartości, aby nowe wagi były zawsze nieujemne; w tym celu dla każdej krawędzi (v_1,v_2) musi zachodzić:

$$waga(v_1,v_2) + \pi(v_1) - \pi(v_2) \geqslant 0,$$

czyli

$$\pi(v_2) \leqslant \pi(v_1) + waga(v_1,v_2).$$

Zaobserwujmy, że nierówność tę spełnia (jako potencjał) funkcja odległości od wierzchołka s. Możemy ten fakt wykorzystać w następujący sposób: początkowo potencjał każdego wierzchołka przyjmiemy równy 0, podczas wyszukiwania nowej ścieżki powiększającej będziemy obliczać odległość każdego wierzchołka od źródła s i podczas wyszukiwania następnej ścieżki powiększającej użyjemy tych odległości jako nowej funkcji potencjału. Należy sobie jeszcze zadać pytanie, czy podczas odwracania krawędzi funkcja potencjału nie straci swoich właściwości. Okazuje się, że każda krawędź (v_1, v_2) leżąca na najkrótszej ścieżce powiększającej spełnia następującą równość:

$$\pi(v_2) = \pi(v_1) + waga(v_1, v_2)$$

zatem zarówno przed jak i po odwróceniu tej krawędzi jej waga będzie wynosić 0. Zwrot pozostałych krawędzi nie ulega zmianom. Ostatecznie nowe wagi wszystkich krawędzi będą nieujemne, co pozwala nam na użycie algorytmu Dijkstry do wyszukiwania najkrótszych ścieżek powiększających.

Testy

Zadanie testowane było na 15 zestawach danych testowych, które zawierały łącznie 20 testów. W poniższej tabeli n oznacza liczbę szkół w teście, a k — przedział w jakim mieszczą się wszystkie stopnie wierzchołków reprezentujących szkoły w grafie (obrazuje to gęstość grafu).

Nazwa	**n**	**k**	**Opis**
szk1a.in	3	$1 \div 2$	bardzo mały test z odpowiedzią NIE
szk1b.in	10	$2 \div 10$	bardzo mały test
szk3.in	5	$1 \div 4$	test losowy
szk4.in	16	$1 \div 15$	test losowy
szk5a.in	12	$1 \div 12$	test losowy
szk5b.in	200	$10 \div 10$	tylko 10 dozwolonych numerów szkół
szk7.in	15	$1 \div 3$	test losowy
szk8.in	60	$1 \div 3$	test losowy
szk9.in	120	$1 \div 2$	test losowy
szk10a.in	30	$1 \div 30$	test losowy
szk10b.in	70	$10 \div 20$	test losowy z odpowiedzią NIE
szk12.in	100	$10 \div 20$	test losowy
szk13.in	200	$2 \div 200$	test losowy
szk14a.in	200	$200 \div 200$	graf pełny dwudzielny
szk14b.in	200	$1 \div 10$	test losowy z odpowiedzią NIE
szk16.in	200	$37 \div 200$	wszystkie szkoły mają ten sam numer początkowy
szk17.in	200	$1 \div 10$	test losowy
szk18a.in	200	$100 \div 100$	test losowy
szk18b.in	200	$199 \div 199$	jeden numer nie jest dopuszczalny dla żadnej ze szkół
szk20.in	200	$50 \div 100$	graf o specyficznej strukturze