

## Zadanie 5: Zadania grafowe

Rozwiązanie każdego z wariantów zadania polega na:

- zaproponowaniu efektywnego czasowo i pamięciowo rozwiązania zdefiniowanego problemu,
- implementacji rozwiązania,
- określeniu pesymistycznej złożoności czasowej i pamięciowej rozwiązania.

### Wariant A „Potop”

Na prostokątnej szachownicy składającej się z  $n \times m$  pól ustawiono  $n \cdot m$  prostopadłościanów - na każdym polu jeden prostopadłościan. Podstawa każdego prostopadłościanu pokrywa się z jednym polem szachownicy i ma powierzchnię jednego metra kwadratowego. Prostopadłościany na sąsiednich polach ściśle przylegają do siebie i nie tworzą żadnych szczelin. Na tę konstrukcję spadł ulewny deszcz. W niektórych miejscach utworzyły się zastoiska wody. Zaproponuj możliwie najefektywniejszy algorytm, który:

- wczyta z pliku tekstowego rozmiary szachownicy oraz wysokości prostopadłościanów ustawionych na poszczególnych polach,
- obliczy maksymalną objętość wody, która po deszczu może pozostać w zastoiskach,
- zapisze wyniki w pliku tekstowym.

Wejście:

W pierwszym wierszu pliku wejściowego Pot\_in\_grupa\_nazwisko.txt są zapisane dwie dodatnie liczby całkowite  $n$  i  $m$ ,  $1 \leq n \leq 100$ ,  $1 \leq m \leq 100$ . Są to rozmiary szachownicy. W każdym z kolejnych  $n$  wierszy znajduje się  $m$  liczb całkowitych z przedziału  $[1..10000]$ :  $i$ -ta liczba w  $j$ -tym wierszu jest wyrażoną w metrach wysokością prostopadłościanu stojącego na przecięciu  $i$ -tej kolumny i  $j$ -tego wiersza szachownicy.

Wyjście:

Twój program powinien zapisać w pierwszym i jedynym wierszu pliku wyjściowego Pot\_out\_grupa\_nazwisko.txt jedną liczbę całkowitą równą maksymalnej objętości wody (wyrażoną w metrach sześciennych), która może zebrać się w zastoiskach konstrukcji.

#### Przykład

Pliku wejściowy Pot\_in\_grupa\_nazwisko.txt:

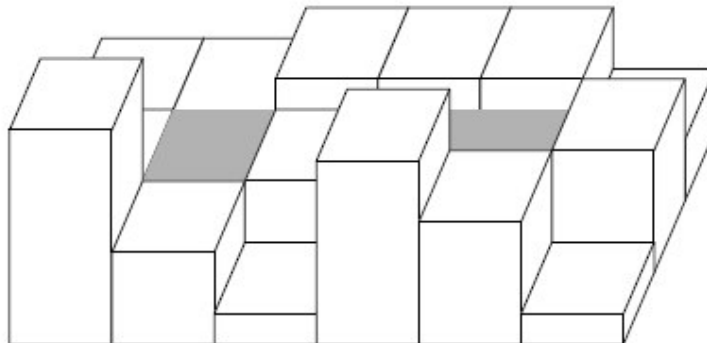
```
3 6
3 3 4 4 4 2
3 1 3 2 1 4
7 3 1 6 4 1
```

Plik wyjściowy Pot\_out\_grupa\_nazwisko.txt:

5

J.Koszelew

Poniższy rysunek przedstawia szachownice po deszczu.



### Wariant B „Wycieczka”

Pewna agencja turystyczna chce wejść na rynek oferując zwiedzanie miasta autobusem-kabrioletem. Należy zbudować siedzibę firmy, w której będzie się zaczynało i kończyło zwiedzanie. Trasa zwiedzania musi przechodzić wszystkimi ulicami miasta, w przeciwnym przypadku turyści mogliby podejrzewać, że nie zobaczyli czegoś bardzo interesującego. Ulice nie muszą być proste i mogą przebiegać tunelami lub wiaduktami. Wszystkie ulice są dwukierunkowe. Każda ulica łączy dwa skrzyżowania. Z każdego skrzyżowania w czterech kierunkach wychodzą ulice. Może się zdarzyć, że dwa skrzyżowania są połączone więcej niż jedną ulicą. Na ulicach nie wolno zawracać, ale można to robić na skrzyżowaniach. Ponadto wiadomo, że z każdego skrzyżowania da się dojechać do każdego innego. Przy każdej ulicy, dokładnie w połowie drogi pomiędzy skrzyżowaniami, które łączy ulica, znajduje się szczególnie godna podziwu atrakcja turystyczna (np. piękny widok, pomnik lub inny zabytek), wywierająca na zwiedzających „wrażenie” określone nieujemną liczbą całkowitą. Siedziba agencji powinna znajdować się przy jednej z takich atrakcji. Przy doborze trasy zwiedzania należy brać pod uwagę zainteresowanie turystów, które może się zmieniać w trakcie zwiedzania. Przejechanie autobusem jednego kilometra powoduje spadek zainteresowania o jeden. Przejechanie po raz pierwszy obok danej atrakcji turystycznej zwiększa zainteresowanie turystów, o liczbę określającą wrażenie, jakie robi atrakcja. Początkowo poziom zainteresowania turystów jest równy wrażeniu, jakie robi atrakcja, przy której znajduje się siedziba agencji. Zainteresowanie turystów nie może w trakcie wycieczki nigdy spaść poniżej zera. Zaproponuj możliwie najefektywniejszy algorytm, który:

- wczyta opis miasta z pliku tekstowego,
- znajdzie trasę spełniającą podane wymagania lub stwierdzi, że taka trasa nie istnieje,
- zapisze wynik do pliku tekstowego

Wejście:

W pierwszym wierszu pliku tekstowego `Wyc_in_grupa_nazwisko.txt` znajduje się jedna liczba całkowita  $n$  określająca liczbę skrzyżowań,  $1 < n \leq 10000$ . Skrzyżowania są ponumerowane od 1 do  $n$ , a ulice są ponumerowane od 1 do  $2n$ . Kolejnych  $2n$  wierszy opisuje ulice -  $(i+1)$ -szy wiersz w pliku opisuje ulicę o numerze  $i$ . W każdym wierszu znajdują się

J.Koszelew

cztery liczby całkowite  $a, b, l, s$  oddzielone pojedynczymi odstępami. Liczby  $a$  i  $b$  to numery skrzyżowań, które łączy dana ulica,  $1 \leq a, b \leq n$ ,  $a \neq b$ . Liczba  $l$  jest parzystą liczbą całkowitą będącą długością ulicy w kilometrach,  $2 \leq l \leq 1\ 000$ . Atrakcja turystyczna położona przy danej ulicy robi wrażenie określone liczbą  $s$ ,  $0 \leq s \leq 1\ 000$ .

Wyjście:

Pierwszy wiersz pliku tekstowego Wyc\_out\_grupa\_nazwisko.txt powinien zawierać jedno słowo TAK, jeżeli istnieje taka trasa, lub NIE, w przeciwnym przypadku. Jeśli odpowiedź jest pozytywna to kolejne wiersze powinny opisywać przykładową trasę. Drugi wiersz powinien zawierać dokładnie jedną liczbę całkowitą  $k$  równą liczbie skrzyżowań występujących na trasie zwiedzania. (Pamiętaj, że ulica, przy której ma znajdować się siedziba agencji łączy pierwsze i ostatnie skrzyżowanie). Oznaczmy przez  $s_i$  (dla  $i = 1, 2, \dots, k$ ) numer ulicy, która podczas zwiedzania dojeżdża się do  $i$ -tego (w kolejności zwiedzania) skrzyżowania. Kolejny wiersz powinien zawierać dwie liczby całkowite  $s_1$  i  $d$  równe odpowiednio numerowi ulicy, przy której należy zbudować siedzibę agencji oraz numerowi pierwszego skrzyżowania, przez które prowadzi trasa zwiedzania. Kolejne  $k-1$  wierszy powinno zawierać po jednej liczbie całkowitej, odpowiednio  $s_2, s_3, \dots, s_k$ .

Przykład

Plik wejściowy Wyc\_in\_grupa\_nazwisko.txt:

```
4
1 2 4 6
2 4 2 4
3 2 4 2
4 3 10 8
2 1 8 7
4 3 2 1
1 4 2 6
3 1 4 5
```

Plik wyjściowy Wyc\_out\_grupa\_nazwisko.txt

```
TAK
8
5 2
2
6
3
1
8
4
7
```

**Wariant C „Złoto”**

ASD-Landia słynie z bogatych złóż złota, dlatego przez długie lata kwitła sprzedaż tego kruszcu do sąsiedniego królestwa, GRAF-Landii. Niestety powiększająca się ostatnio dziura budżetowa zmusiła króla GRAF-Landii do wprowadzenia zaporowych ceł na metale i

J.Koszelew

minerały. Handlarze przekraczający granicę muszą zapłacić 50% wartości przewożonego ładunku. ASD-Landzkim kupcom grozi bankructwo. Na szczęście alchemicy z tego kraju opracowali sposoby pozwalające zamieniać pewne metale w inne. Pomysł kupców polega na tym, aby z pomocą alchemików zamieniać złoto w pewien tani metal, a następnie, po przewiezieniu go przez granicę i zapłaceniu niewielkiego cła, znowu otrzymywać z niego złoto. Niestety alchemicy nie znaleźli sposobu na zamianę dowolnego metalu w dowolny inny. Może się więc zdarzyć, że proces otrzymania danego metalu ze złota musi przebiegać wielostopniowo i że na każdym etapie uzyskiwany będzie inny metal. Alchemicy każą sobie słono płacić za swoje usługi i dla każdego znanego sobie procesu zamiany metalu  $A$  w metal  $B$  wyznaczyli cenę za przemianę 1 kg surowca. Handlarze zastanawiają się, w jakiej postaci należy przewozić złoto przez granicę oraz jaki ciąg procesów alchemicznych należy zastosować, aby zyski były możliwie największe. Zaproponuj możliwie najefektywniejszy algorytm, który:

- na podstawie cen wszystkich metali, a także ceny przemian oferowanych przez alchemików,
  - wyznaczy taki ciąg metali  $m_0, m_1, \dots, m_k$ , że:
    - $m_0 = m_k$  to złoto
    - dla każdego  $i = 1..k$  alchemicy potrafią otrzymać metal  $m_i$  z metalu  $m_{i-1}$  oraz
    - koszt wykonania całego ciągu procesów alchemicznych dla 1 kg złota powiększony o płacone na granicy cło (50% ceny 1 kg najtańszego z metali  $m_i$ , dla  $i = 0 \dots k$ ) jest najmniejszy z możliwych.
- Zakładamy, że podczas procesów alchemicznych waga metali nie zmienia się.
- wypisz koszt wykonania wyznaczonego ciągu procesów alchemicznych powiększony o płacone na granicy cło.

Wejście

W pierwszym wierszu pliku ZL\_in\_grupa\_nazwisko.txt znajduje się jedna dodatnia liczba całkowita  $n$  oznaczająca liczbę rodzajów metali,  $1 \leq n \leq 5000$ . W wierszu o numerze  $k+1$ , dla  $1 \leq k \leq n$ , znajduje się nieujemna parzysta liczba całkowita  $p_k$  - cena 1 kg metalu oznaczonego numerem  $k$ ,  $0 \leq p_k \leq 10^9$ . Przyjmujemy, że złoto ma numer 1. W wierszu o numerze  $n+2$  znajduje się jedna nieujemna liczba całkowita  $m$  równa liczbie procesów przemiany znanych alchemikom,  $0 \leq m \leq 100000$ . W każdym z kolejnych  $m$  wierszy znajdują się po trzy liczby naturalne, pooddzielane pojedynczymi odstępami, opisujące kolejne procesy przemiany. Trójka liczb  $a; b; c$  oznacza, że alchemicy potrafią z metalu o numerze  $a$  otrzymywać metal o numerze  $b$  i za zamianę 1 kg surowca każą sobie płacić  $c$  asdółów,  $1 \leq a, b \leq n; 0 \leq c \leq 10\,000$ . Uporządkowana para liczb  $a$  i  $b$  może się pojawić w danych co najwyżej jeden raz.

Wyjście:

W pierwszym wierszu pliku wyjściowego ZL\_out\_grupa\_nazwisko.txt powinna zostać wypisana jedna liczba całkowita – koszt wykonania wyznaczonego ciągu procesów alchemicznych powiększony o płacone na granicy cło.

J.Koszelew

#### Przykład

Plik wejściowy ZL\_in\_grupa\_nazwisko.txt:

4

200

100

40

2

6

1 2 10

1 3 5

2 1 25

3 2 10

3 4 5

4 1 50

Plik wyjściowy ZL\_out\_grupa\_nazwisko.txt

60

#### **Wariant D „Księżniczka”**

Król ASD-Landii zamierza wydać za mąż swą urodziwą córkę, księżniczkę ASDellę. Zapytał ją, jakiego męża chciałaby mieć. Księżniczka odpowiedziała, że jej przyszły małżonek powinien być mądry, a także ani skąpy, ani rozrzutny. Zamyślił się król nad tym, jakim to próbom powinni być poddani kandydaci na męża, aby mógł wybrać dla swej córki najlepszego z nich. Po długich dumaniach stwierdził, że najlepiej posłuży do tego zamek, który kazał był wybudować ku uciesze mieszkańców ASD-Landii. Zamek składa się z dużej liczby komnat, w których zgromadzono bogactwa królestwa. Komnaty te, połączone korytarzami, mogą być zwiedzane przez poddanych w celu podziwiania wystawionych w nich wspaniałości. Za zwiedzenie komnaty trzeba uiścić pewną liczbę asdół (asdol jest jednostką pieniężną ASD-Landii). Zwiedzanie zamku rozpoczyna się od komnaty wejściowej. Król wręczył sakiewkę każdemu kandydatowi na męża księżniczki. W każdej sakiewce była taka sama liczba asdółów. Poprosił każdego kandydata, aby ten wybrał taką drogę, która pozwoli, poczynając od komnaty wejściowej, odwiedzić pewną liczbę komnat zamku oraz zakończyć zwiedzanie w komnacie, w której przebywa księżniczka, i wydać przy tym dokładnie kwotę, jaka była w sakiewce. Rozrzutni kandydaci, którzy wydawali po drodze zbyt dużo, nie docierali do komnaty księżniczki, skąpi natomiast zjawiali się z niepustą sakiewką i księżniczka wyprawiała ich w dalszą drogę celem opróżnienia sakiewki. Niestety do dziś żadnemu z kandydatów nie udało się sprostać zadaniu króla, a księżniczka ASDella wciąż z utęsknieniem czeka na swój ideał. Zaproponuj możliwie najefektywniejszy algorytm, który:

- wczyta z pliku tekstowego opis zamku, numer komnaty, w której znajduje się księżniczka i kwotę w sakiewce,
- wyznaczy ciąg komnat, przez które należy kolejno przejść, aby dojść z komnaty wejściowej do komnaty, w której znajduje się księżniczka i wydać dokładnie całą zawartość sakiewki,
- zapisze znaną drogę w pliku tekstowym.

J.Koszelew

Zakładamy, że droga zawsze istnieje. Jeśli istnieje wiele takich dróg, to Twój algorytm powinien wyznaczyć dowolną z nich.

Wejście:

W pierwszym wierszu pliku tekstowego KS\_in\_grupa\_nazwisko.txt zapisanych jest pięć dodatnich liczb całkowitych  $n, m, w, k, s$ ,  $1 \leq n \leq 100$ ,  $1 \leq m \leq 64950$ ,  $1 \leq w, k \leq n$ ,  $1 \leq s \leq 1000$ , pooddzielanych pojedynczymi odstępami. Liczba  $n$  jest równa liczbie komnat, a  $m$  liczbie korytarzy. Komnaty są ponumerowane od 1 do  $n$ . Liczba  $w$  jest numerem komnaty wejściowej, a  $k$  numerem komnaty, w której znajduje się książeczka. Liczba  $s$  określa liczbę asdółw w sakiewce. W drugim wierszu zapisanych jest  $n$  dodatnich liczb całkowitych  $o_1, o_2, \dots, o_n$ ,  $1 \leq o_i \leq 1000$ , pooddzielanych pojedynczymi odstępami. Liczba  $o_i$  jest równa opłacie za (każdorazowy) wstęp do komnaty nr  $i$ . W kolejnych  $m$  wierszach zapisane są po dwie dodatnie liczby całkowite  $x, y$ ,  $x \leq y$ ,  $1 \leq x, y \leq n$ , oddzielone pojedynczym odstępem. Każda taka para  $x, y$  określa, że komnaty o numerach  $x$  i  $y$  są połączone korytarzem.

Wyjście:

Twój program powinien w pierwszym (i jedynym) wierszu pliku KS\_out\_grupa\_nazwisko.txt zapisać ciąg dodatnich liczb całkowitych, pooddzielanych pojedynczymi odstępami, określający numery kolejnych komnat, przez które należy przejść, aby dojść z komnaty wejściowej do komnaty, w której znajduje się książeczka, i wydać dokładnie całą zawartość sakiewki.

Przykład

Plik wejściowy KS\_in\_grupa\_nazwisko.txt:

5 6 3 4 9

1 2 3 4 5

2 4

5 4

1 5

1 2

2 3

3 1

Plik wyjściowy KS\_out\_grupa\_nazwisko.txt:

3 2 4

J.Koszelew

