

## **Zadanie 6: Strategia powrotów i algorytmy przybliżone**

Rozwiązanie każdego z wariantów zadania polega na:

- zaproponowaniu i zaimplantowaniu efektywnego czasowo i pamięciowo rozwiązania dokładnego zdefiniowanego problemu,
- zaproponowaniu i zaimplantowaniu efektywnego czasowo i pamięciowo rozwiązania przybliżonego zdefiniowanego problemu,
- ustaleniu przypadków danych, dla których rozwiązanie przybliżone obarczone jest możliwie największym błędem.

### **Wariant A ("Sesja = Egzaminy")**

Dany jest zbiór przedmiotów, które mogą wybierać studenci. Planując terminy egzaminów z tych przedmiotów, należy pamiętać, aby nie zaplanować jednocześnie dwóch egzaminów, jeżeli istnieje student, który wybrał oba egzaminowane przedmioty. Choć zawsze istnieje niekonfliktowe rozwiązanie, w którym każdy egzamin odbywa się w innym terminie, to jednak dążymy do tego, aby harmonogram egzaminów zawierał możliwie najmniejszą liczbę terminów. Zaproponuj algorytm, który na podstawie informacji na temat przedmiotów zadawanych przez studentów wybierze możliwie najmniejszą liczbę terminów egzaminów zapewniających brak konfliktów w harmonogramie sesji (żaden student nie ma zaplanowanych dwóch różnych egzaminów w tym samym terminie).

Plik wejściowy: Plik tekstowy o nazwie `Sesja_nazwisko_grupa_in.txt`. Plik ma następujący format: W pierwszej linii pliku znajdują się dwie liczby całkowite  $n$  i  $m$  oddzielone pojedynczą spacją ( $1 < n, m \leq 200$ ). Liczba  $n$  oznacza liczbę przedmiotów, które kończą się egzaminem,  $m$  liczbę studentów, którzy przystępują do egzaminów. W kolejnych  $m$  liniijkach znajduje się informacja opisująca numery przedmiotów, z których poszczególni studenci zdają egzaminy.  $i$ -ta liniijka ( $1 \leq i \leq m$ ) zawiera co najwyżej  $n$  liczb określających numery przedmiotów zdawane przez studenta o numerze  $i$ , oddzielone pojedynczą spacją.

Plik wyjściowy: Plik tekstowy o nazwie `Sesja_nazwisko_grupa_out.txt` ma następujący format: W pierwszej linii pliku znajduje się liczba  $min$  - minimalna terminów egzaminów, które mogą być określone tak, że każdy student może zdać wybrane przez siebie egzaminy w niekonfliktowych terminach. W kolejnych  $min$  liniijkach znajduje się informacja opisująca numery przedmiotów, które mogą być zdawane w poszczególnych terminach.  $i$ -ta liniijka zawiera co najwyżej  $n$  liczb całkowitych oddzielonych pojedynczą

J.Koszelew

spacją i oznaczających numery przedmiotów, które mogą być zdawane w terminie o numerze  $i$ .

### Przykład

Sesja\_nazwisko\_grupa\_in.txt:

5 3

1 3 4

2 5

4 5

Sesja\_nazwisko\_grupa\_out.txt

3

1 5

2 4

3

### **Wariant B** ("Kasia zna Wojtkę=Wojtek zna Kasię")

W pewnej grupie studentów znane są wszystkie relacje znajomości, które są oczywiście obustronne (tzn. jeżeli student  $A$  zna studenta  $B$  to student  $B$  przyznaje się do znajomości ze studentem  $A$ ). Zaproponuj algorytm, który określa skład jednej, maksymalnej co do liczby studentów podgrupy, w której wszyscy studenci znają się wzajemnie.

Plik wejściowy: Plik tekstowy o nazwie Kasia\_nazwisko\_grupa\_in.txt ma następujący format: W pierwszej linijce pliku znajduje się liczba całkowita  $n$  ( $0 < n \leq 200$ ) określająca liczbę studentów w grupie. W kolejnych  $n$  linijkach pliku znajduje się co najwyżej  $n-1$  liczb oddzielonych pojedynczą spacją i oznaczających znajomości danego studenta. W  $i$ -tej linijce znajdują się numery studentów, który zna (oprócz siebie oczywiście) student o numerze  $i$ .

Plik wyjściowy: Plik tekstowy o nazwie Kasia\_nazwisko\_grupa\_out.txt ma następujący format: W pierwszej linijce pliku znajduje się jedna liczba całkowita  $max$  oznaczająca rozmiar największej podgrupy wzajemnie znających się studentów. W drugiej linijce

J.Koszelew

znajduje się *max* liczb całkowitych oddzielonych pojedynczą spacją i oznaczających numery studentów, którzy należą do ustalonej maksymalnej podgrupy.

#### Przykład

Plik Kasia\_nazwisko\_grupa\_in.txt:

5

2 3 4

1 3 4 5

1 2 4 5

1 2 3 5

2 3 4

Plik Kasia\_nazwisko\_grupa\_out.txt:

4

2 3 4 5

#### **Wariant C („Sieć przewodowa=Sieć transformatorów”)**

W dużym nowym osiedlu planowane jest postawienie transformatorów obsługujących osiedlową sieć przesyłu energii elektrycznej. Ustalono już możliwe do zrealizowania lokalizacje transformatorów i projekt linii przesyłu łączących te transformatory. Należy jeszcze jednak, ze zbioru ustalonych lokalizacji, wybrać te, które zapewnią przesył energii każdą linią sieci. Przy czym należy zadbać o to, aby wybranych lokalizacji zapewniających przesył na każdym łączu sieci było jak najmniej.

Zaproponuj algorytm, który na podstawie grafu sieci ustali najmniejszy podzbiór lokalizacji transformatorów, zapewniający przesył energii w każdym łączu sieci.

Plik wejściowy: Plik tekstowy o nazwie Siec\_nazwisko\_grupa\_in.txt ma następujący format: W pierwszej linijce pliku znajduje się liczba całkowita  $n$  ( $0 < n \leq 200$ ) określająca liczbę transformatorów. W kolejnych  $n$  liniijkach pliku znajduje się co najwyżej  $n-1$  liczb oddzielonych pojedynczą spacją i oznaczających linie przesyłu obsługiwane przez dany

J.Koszelew

transformator. W  $i$ -tej linii znajdują się numery transformatorów, z którymi jest połączony liniami przesyłu transformator o numerze  $i$ .

Plik wyjściowy: Plik tekstowy o nazwie `Siec_nazwisko_grupa_out.txt` ma następujący format: W pierwszej linii pliku znajduje się jedna liczba całkowita  $min$  oznaczająca rozmiar podzbioru wybranych lokalizacji transformatorów. W drugiej linii znajduje się  $min$  liczb całkowitych oddzielonych pojedynczą spacją i oznaczających numery lokalizacji, które należą do tego minimalnego podzbioru.

#### Przykład

`Siec_nazwisko_grupa_in.txt`

7

3 4

4 5 7

1

1 2 7

2 7

7

2 4 5 6

`Siec_nazwisko_grupa_out.txt`

3

2 4 7

#### **Wariant D („Szybka sieć=Multicast Routing”)**

W pewnym mieście powstanie nowa sieć oparta o zbiór komputerów osobistych  $K$  i routerów  $R$ . Znane są koszty przygotowania możliwych połączeń między wierzchołkami zbioru  $K \cup R$  w projektowanej sieci. Połączenia mogą łączyć komputer z komputerem, komputer z routerem albo dwa routery i są nieskierowane. Zaproponuj algorytm, który wybiera taki podzbiór połączeń w sieci, który zapewni komunikację pomiędzy każdą parę

J.Koszelew

komputerów ze zbioru  $K$  o możliwie najmniejszej sumie kosztów wybranych połączeń. Wybrane krawędzie mogą łączyć komputer z komputerem, komputer z routerem albo dwa routery.

Plik wejściowy: Plik tekstowy o nazwie MC\_nazwisko\_grupa\_in.txt ma następujący format: W pierwszej linijce pliku znajdują się dwie liczby całkowite  $n$  i  $k$  ( $0 < n, k \leq 200$ ) określające odpowiednio  $n$  - liczbę routerów i komputerów,  $k$  - liczbę komputerów. W kolejnej linijce znajduje się  $k$  liczb całkowitych z zakresu od 1 do  $n$ , oddzielonych spacją, oznaczających numery komputerów. W kolejnych  $n$  liniach znajdują się wiersze tablicy wag połączeń w sieci. Zakładamy, że wagi połączeń to liczby całkowite  $w_{ij}$ ,  $0 \leq w_{ij} \leq 200$ ;  $w_{ii} = 0$ ;  $w_{ij} = 300$ , gdy między wierzchołkami  $i$  oraz  $j$  nie ma możliwości bezpośredniego połączenia.

Plik wyjściowy: Plik tekstowy o nazwie MC\_nazwisko\_grupa\_out.txt ma następujący format: W pierwszej linijce pliku znajduje się jedna liczba całkowita  $p$  oznaczająca liczbę wybranych połączeń. W kolejnych  $p$  liniach zapisane są pary liczb oddzielone spacją oznaczające wybrane połączenie - wierzchołki należące do połączenia.

#### Przykład

MC\_nazwisko\_grupa\_in.txt

9 4

1 3 7 9

0 2 300 3 300 300 300 300 300

2 0 2 300 1 300 300 300 300

300 2 0 300 300 2 300 300 300

3 300 300 0 2 300 2 300 300

300 1 300 2 0 2 300 2 300

300 300 2 300 2 0 300 300 3

300 300 300 2 300 300 2 1 300

300 300 300 300 2 300 1 0 2

J.Koszelew

300 300 300 300 300 3 300 2 0

MC\_nazwisko\_grupa\_out.txt

8

1 2

2 3

2 5

4 5

5 6

5 8

7 8

8 9