

# LMAD - Algorytmy grafowe (pakiet NetworkX)

## A Zadania na ćwiczenia

**Zadanie A.1.** Czechy pracują nad nowatorską metodą transportowania piwa za pomocą rurociągu. Jako pierwszy ma powstać piwociąg łączący miejscowość Královec z  $n$  wytypowanymi miastami. Z przyczyn bezpieczeństwa, piwociąg musi zostać wybudowany wzdłuż istniejących dróg. Ponadto inżynierowie oszacowali koszt budowy piwociągu wzdłuż każdej drogi. Ponieważ rząd czeski zмага się z galopującą inflacją, budżet inwestycji jest mocno ograniczony. Twoim zadaniem jest sprawdzić, czy inwestycja może zostać zrealizowana. Jeśli jest to możliwe, podaj przykład rozplanowania piwociągu realizującego minimalny koszt budowy.

Na wejściu znajdują się trzy liczby  $n$ ,  $m$ ,  $b$  oznaczające odpowiednio liczbę miast (wraz z Královcem), liczbę dróg pomiędzy miastami oraz przewidziany budżet inwestycji (w miliardach CZK). W kolejnym wierszu znajduje się lista miast, do których ma zostać doprowadzony piwociąg (pierwszy na liście jest Královec). W kolejnych  $m$  wierszach podane są istniejące drogi (czyli pary miast) oraz koszt budowy piwociągu wzdłuż nich (w miliardach CZK).

Na wyjściu należy podać czy da się przeprowadzić inwestycję. Jeśli tak, należy podać minimalny koszt (w miliardach CZK) oraz przykład jej realizacji (czyli listę dróg, wzdłuż których będzie budowany piwociąg).

**Przykład:**

Wejście:

5 6 13

A, B, C, D, E

A B 5

A E 7

B D 9

B E 1

C D 4

D E 2

Wyjście:

TAK

12

[('A', 'B'), ('B', 'E'), ('C', 'D'), ('D', 'E')]

**Zadanie A.2.** Wygeneruj wszystkie drzewa na zbiorze wierzchołków  $\{0, 1, \dots, n-1\}$  ( $n \geq 3$ ), w których wierzchołek 0 ma stopień  $k$ , a wierzchołek 1 ma stopień  $n-k$ .

Na wejściu podane są dwie liczby  $n$  i  $k$  każda w osobnej linii. Na wyjściu należy podać listy krawędzi wszystkich wygenerowanych drzew, każdą w osobnej linii. Każda lista krawędzi powinna być uporządkowana leksykograficznie. Ponadto, lista list krawędzi też powinna być uporządkowana leksykograficznie.

*Wskazówka:* W tym zadaniu mogą używać Państwo funkcji *distinct\_permutations* z pakietu *more\_itertools*.

**Zadanie A.3.** W ogrodzie znajduje się  $n$  placów z fontannami, które są połączone alejami. Wzdłuż każdej alei rosną krzewy, które musi przyciąć ogrodnik. Ogrodnik chce wykonać pracę możliwie efektywnie, więc chce przejść każdą aleję dokładnie raz. Ponadto, jeśli to możliwe, chciałby zakończyć pracę na tym samym placu, na którym zaczynał. Określ, czy jest to możliwe i wygeneruj przykładową trasę ogrodnika.

Na wejściu podane są w kolejnych liniach liczba placów  $n$ , liczba łączących je alei  $m$ , uporządkowaną listę  $n$  placów, a następnie listę  $m$  par placów oznaczających, że pomiędzy daną parą placów znajduje się aleja (każda taka para jest umieszczana w osobnej linii). Zakładamy, że plac, na którym ogrodnik zaczyna pracę jest podany jako pierwszy na liście  $n$  miejsc. Na wyjściu należy podać, czy możliwe jest stworzenie poprawnej trasy w postaci jednego z trzech komunikatów:

- Można utworzyć trasę i powrócić na plac startowy.
- Można utworzyć trasę, ale bez powrotu na plac startowy.
- Nie można utworzyć trasy.

Ponadto, jeśli można utworzyć trasę (z lub bez powrotu na plac startowy), to należy w drugiej linii podać listę kolejnych alei, wzdłuż których pracuje ogrodnik (jako par placów).

**Przykład**

Wejście

5

8

A B C D E F

A B

A C  
A D  
A E  
B C  
B E  
B F  
D F

Wyjście

Można utworzyć trasę i powrócić na plac startowy

[('A', 'E'), ('E', 'B'), ('B', 'F'), ('F', 'D'), ('D', 'A'), ('A', 'C'), ('C', 'B'), ('B', 'A')]

**Zadanie A.4.** Ania zrobiła listę  $n$  miejsc w Polsce, które chce odwiedzić. Co weekend chce odwiedzać jedno z nich (każde dokładnie raz). Miejsca są połączone siecią  $m$  dróg, przy czym drogi są zawsze dwukierunkowe i przebycie każdej z nich zajmuje tyle samo czasu. Plan Ani zakłada, że będzie odwiedzać miejsca w kolejności od tego, które jest najdalej (czyli wymagające w najlepszym przypadku przebycia największej liczby dróg) do tego, które jest najbliżej, a jeśli kilka z nich jest w tej samej odległości, to najpierw wybierze się do tego, które jest wcześniej na liście.

Napisz program, który pomoże Ani ustalić kolejność odwiedzanych miejsc. Program ma przyjmować jako dane wejściowe liczby  $n$  i  $m$ , uporządkowaną listę  $n$  miejsc oraz listę  $m$  par miejsc oznaczających, że pomiędzy daną parą miejsc znajduje się droga (każda taka para jest umieszczana w osobnej linijce). Zwracana ma być lista miejsc do odwiedzenia w kolejności od pierwszego odwiedzanego miejsca do ostatniego odwiedzanego miejsca. Zakładamy, że Ania mieszka w miejscu podanym jako pierwsze na liście  $n$  miejsc (tego miejsca nie podajemy na wyjściu).

*Uwaga!* Można przyjąć, że lista miejsc będzie zawsze uporządkowana leksykograficznie.

**Przykład:**

Wejście:

5

6

A, B, C, D, E

A B

A E

B D

B E

C D

D E

Wyjście:

[C, D, B, E]

## B Zadania na ćwiczenia - jeśli czas pozwoli

**Zadanie B.1.** Wygeneruj wszystkie drzewa na zbiorze wierzchołków  $\{0, 1, \dots, n-1\}$  ( $n \geq 4$ ), w których wierzchołek  $n-1$  ma stopień  $n-3$ .

Na wejściu podane jest liczba  $n$ . Na wyjściu należy podać listy krawędzi wszystkich wygenerowanych drzew, każdą w osobnej linijce. Każda lista krawędzi powinna być uporządkowana leksykograficznie. Ponadto, lista list krawędzi też powinna być uporządkowana leksykograficznie.

*Wskazówka:* W tym zadaniu mogą używać Państwo funkcji `distinct_permutations` z pakietu `more_itertools`.

## C Zadania do samodzielnej pracy w domu

**Zadanie C.1.** Adam podróżuje tramwajem po mieście. Za każdy przejechany przystanek (poza staartowym) płaci 0,5 zł. Napisz program, który pomoże Adamowi znaleźć najtańszą trasę pomiędzy przystankami startowym  $X$  i końcowym  $Y$ .

Na wejściu podane są w kolejnych wierszach: łączna liczba przystanków w sieci  $n$ , łączna liczba połączeń między przystankami  $m$ , lista wszystkich przystanków, przystanek startowy  $X$ , przystanek końcowy  $Y$  oraz  $m$  par przystanków, między którymi istnieje bezpośrednie połączenie tramwajowe (każda w osobnej linijce). Na wyjściu należy podać przykładową najkrótszą trasę od  $X$  do  $Y$  w postaci listy par przystanków oznaczających kolejno wybierane połączenia oraz w osobnej linijce łączny koszt takiej trasy.

**Zadanie C.2.** Wygeneruj wszystkie drzewa na zbiorze wierzchołków  $\{0, 1, \dots, n-1\}$  ( $n \geq 3$ ), w których istnieją trzy wierzchołki  $u, v, w$  takie, że ich stopnie to  $d(u) = k, d(v) = l, d(w) = m$  dla podanych wartości  $k, l, m$ , które spełniają zależność  $k + l + m = n + 1$ .

Na wejściu podane są cztery liczby  $n, k, l$  i  $m$  każda w osobnej linijce. Na wyjściu należy podać listy krawędzi wszystkich

wygenerowanych drzew, każdą w osobnej linii. Każda lista krawędzi powinna być uporządkowana leksykograficznie. Ponadto, lista list krawędzi też powinna być uporządkowana leksykograficznie.

*Wskazówka:* W tym zadaniu mogą używać Państwo funkcji *distinct\_permutations* z pakietu *more\_itertools*.