

Szablon rozwiązania	egzP3a.py
Złożoność akceptowalna (1.5pkt):	$O(n^2mp)$
Złożoność wzorcowa (+2.5pkt):	$O(nmp)$, gdzie n to max liczba okręgów, m to ilość wyborów, które się odbyły, p to max fundusz na kampanię

W pewnym państwie ordynacja wyborcza mówi, że dla zagwarantowania sprawiedliwości wyborów parlamentarnych, przed każdymi wyborami zmieniają się obszary obejmujące dane okręgi wyborcze (w szczególności oznacza to zmianę ilości mieszkańców wewnątrz danego okręgu). Może zmienić się także sama ilość tych okręgów. Każdy okrąg wyborczy ma przypisaną liczbę mieszkańców obejmowanych przez niego podczas danych wyborów, oraz koszt przeprowadzenia kampanii wyborczej przez Partię X (zakładamy, że gwarantuje ona głos wszystkich mieszkańców tego okręgu, a nie przeprowadzenie kampanii powoduje zero głosów od tych osób). Przed każdymi wyborami szef Partii X wybiera okręgi w których najlepiej będzie przeprowadzić kampanię (tj. tak, aby uzyskać jak największą liczbę głosów) mając do dyspozycji pewien budżet. Przed tegorocznymi wyborami członkowie rady nadzorczej poprosili szefa o stworzenie raportu zawierającego liczbę głosów dotychczas otrzymanych przez partię.

W ramach zadania należy zaimplementować funkcję:

```
def wybory( T )
```

która oblicza całkowitą liczbę głosów otrzymanych przez Partię X, zakładając, że:

1. Ilość okręgów podczas jednych wyborów nigdy nie przekroczyła n , wybory odbyły się dotychczas dokładnie m razy, a maksymalny fundusz na kampanię wyniósł p .
2. Tablica T zawiera wskazania na pierwsze elementy list jednokierunkowych, wyrażających okręgi wyborcze podczas danych wyborów, zrealizowanych jako następująca struktura:

```
class Node:
    def __init__( self, wyborcy, koszt, fundusze ):
        self.next = None
        self.wyborcy = wyborcy
        self.koszt = koszt
        self.fundusze = fundusze
        self.x = None
```

- **wyborcy** to liczba wyborców w danym okręgu
- **koszt** to koszt przeprowadzenia kampanii w danym okręgu
- **fundusze** to łączny budżet przeznaczony na kampanię podczas danych wyborów – wartość ta jest taka sama we wszystkich elementach wewnątrz danej listy tj. podczas danych wyborów
- **next** wskazuje na następny okrąg podczas tych wyborów
- **x** jest pustym polem możliwym do wykorzystania przez studenta

Rozważmy następujące dane:

```
wyb1okr1 = Node(3, 8, 15)    wyb2okr2 = Node(5, 2, 8)
wyb1okr2 = Node(2, 7, 15)    wyb2okr1.next = wyb2okr2
wyb1okr3 = Node(4, 5, 15)    wyb3okr1 = Node(3, 5, 10)
wyb1okr1.next = wyb1okr2     wyb3okr2 = Node(3, 5, 10)
wyb1okr2.next = wyb1okr3     wyb3okr1.next = wyb3okr2
wyb2okr1 = Node(4, 7, 8)     T = [wyb1okr1, wyb2okr1, wyb3okr1]
```

Wywołanie `wybory(T)` powinno zwrócić wynik **18** (Podczas pierwszych wyborów Partia X zdobyła **7** głosów, podczas drugich **5** głosów, a podczas trzecich **6** głosów)

Szablon rozwiązania

egzP3b.py

Złożoność akceptowalna (1.5pkt):

$O(nm)$

Złożoność wzorcowa (+2.5pkt):

$O(n \log m)$, gdzie m to liczba miast, a n to liczba połączeń

Ze względu na rosnącą konkurencję i rozwój tanich linii lotniczych, dla tradycyjnych operatorów utrzymywanie wielu połączeń staje się coraz trudniejsze. Na zebraniu akcjonariuszy Lufthansy zostało postanowione, że muszą zostać usunięte wszystkie połączenia redundantne (tj. takie, które mogą zostać zastąpione innymi połączeniami, np. z przesiadką). Oznacza to, że po wdrożeniu zmian, między każdą parą miast będzie istniało dokładnie jedno unikatowe połączenie. Ustalono, że od tej zasady może istnieć dokładnie jeden wyjątek (tj. dokładnie jedno połączenie może pozostać redundantne). Mimo minimalizacji ilości połączeń, Lufthansa chce jednak zadbać, aby lotów odbywało się jak najwięcej (Jednak zakładamy, że nie zostaną dodane żadne nowe połączenia, ani też nie zwiększy się przepustowość na istniejących). Aktualna mapa połączeń ma postać nieskierowanego grafu ważonego $G = (V, E)$ gdzie wierzchołki to lotniska, a krawędzie to istniejące połączenia (a ich wagi oznaczają aktualną liczbę lotów odbywających się na każdej relacji w ciągu tygodnia). Proszę obliczyć o ile mniej lotów w skali tygodnia będzie wykonywała Lufthansa po wdrożeniu ustaleń.

W ramach zadania należy zaimplementować funkcję:

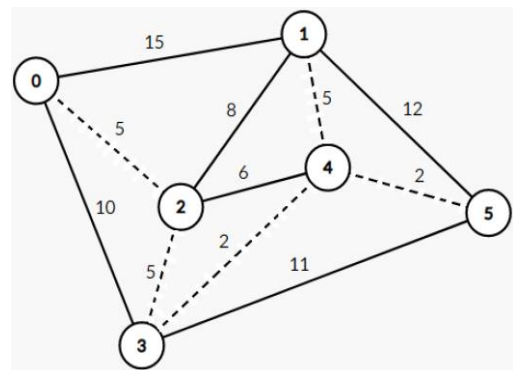
```
def lufthansa( G )
```

która oblicza ile mniej tygodniowo lotów będzie wykonywała Lufthansa, zakładając, że:

1. Graf ważony G jest wyrażony jako lista sąsiedztwa, czyli dla każdego połączenia między punktami u oraz v o wadze w , lista $G[u]$ będzie zawierała krotkę (v, w) oraz lista $G[v]$ będzie zawierała krotkę (u, w)

Rozważmy następujące dane:

```
G = [
    [(1, 15), (2, 5), (3, 10)],
    [(0, 15), (2, 8), (4, 5), (5, 12)],
    [(0, 5), (1, 8), (3, 5), (4, 6)],
    [(0, 10), (2, 5), (4, 2), (5, 11)],
    [(1, 5), (2, 6), (3, 2), (5, 2)],
    [(1, 12), (4, 2), (3, 11)]
]
```



Wywołanie `lufthansa(G)` powinno zwrócić wynik **19** (Jest to minimalna liczba lotów, tj. sum wag usuniętych krawędzi, po wdrożeniu ustaleń z zebrania akcjonariuszy. Usuwamy krawędzie **0-2**, **2-3**, **3-4**, **4-5**, oraz **1-4**)

Podpowiedź. Jak zmieniłyby się warunki zadania, jeżeli nie występowałby wyjątek?