

# Algorytmy i struktury danych

## Laboratorium 6

mgr inż. Andrii Shekhovtsov

16 grudnia 2023

### 1 Zasady oceniania

Ocena za laboratorium zależy od liczby dobrze zrobionych zadań, skala ocen jest pokazana w Tabeli 1. W niektórych przypadkach mogą być uwzględnione zadania rozwiązywane częściowo.

Tabela 1: Skala ocen.	
Liczba dobrze zrobionych zadań	Ocena
1	3.0
2	4.0
4	5.0

Wykonane zadania proszę przesłać za pośrednictwem platformy moodle w postaci pliku tekstowego z kodem w Python (plik z rozszerzeniem `.py`). Proszę wrzucić kod ze wszystkich zadań do jednego pliku, rozdzielając poszczególne zadania za pomocą komentarzy.

**UWAGA:** Termin oddania zadania jest ustawiony w systemie moodle. W przypadku nie oddania zadania w terminie, uzyskana ocena będzie zmniejszana o 0,5 za każdy zaczęty tydzień opóźnienia.

**UWAGA:** W przypadku wysłania zadania w formie niezgodnej z opisem w instrukcji prowadzący zastrzega prawo do wystawienia oceny negatywnej za taką pracę. Przykład: wysłanie `.zip` lub `.pdf` tam, gdzie był wymagany plik tekstowy z rozszerzeniem `.py`.

### 2 Zadania do wykonania

1. Zaimplementować funkcje służącą do wczytania grafu nieskierowanego z pliku tekstowego zawierającego informacje o krawędziach grafu. Funkcja powinna przyjmować nazwę pliku i zwracać słownik zawierający informacje o wierzchołkach oraz listę ich sąsiedzi (reprezentacja grafu w postaci listy sąsiedztwa).

Przykładowo plik o następującej zawartości:

```
1 A B
2 A C
3 A D
4 B D
5 D E
6 C E
```

Powinien być wczytany jako słownik:

```
1 {
2   'A': ['B', 'C', 'D'],
3   'B': ['A', 'D'],
4   'C': ['A', 'E'],
5   'D': ['A', 'B', 'E'],
6   'E': ['C', 'D'],
7 }
```

2. Zaimplementować algorytm wyszukiwania najkrótszej ścieżki w grafie za pomocą przeszukiwania grafu wrzesz. Przetestować działanie zaimplementowanego algorytmu na dwóch grafach dostarczonych razem z instrukcją (pliki `small.csv` i `city.csv`).
3. Zaimplementować funkcje służącą do wczytania grafu nieskierowanego ważonego z pliku tekstowego zawierającego informacje o krawędziach grafu oraz ich wagach. Funkcja powinna przyjmować nazwę pliku i zwracać słownik zawierający informacje o wierzchołkach oraz listę ich sąsiedzi (reprezentacja grafu w postaci listy sąsiedztwa).

Przykładowo plik o następującej zawartości:

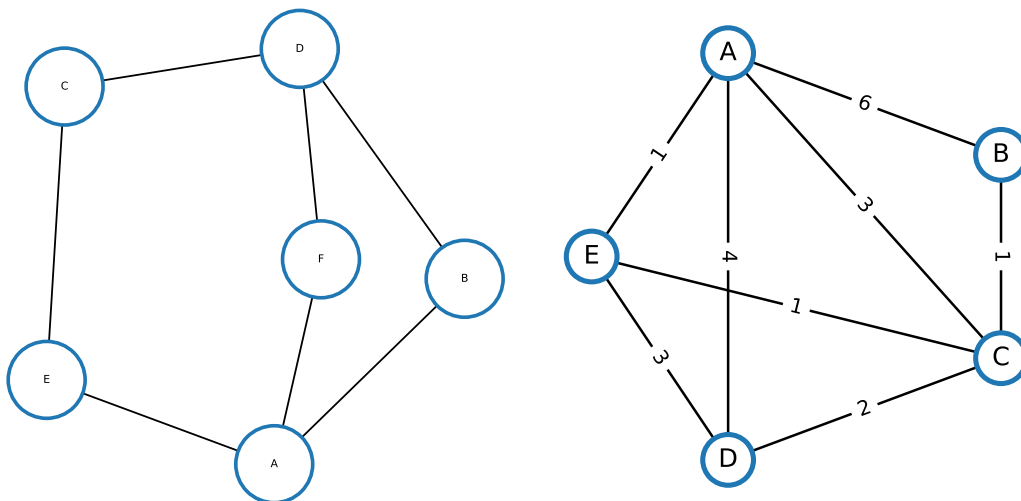
```
1 A B 5
2 A C 3
3 A D 2
4 B D 3
5 D E 1
6 C E 4
```

Powinien być wczytany jako słownik:

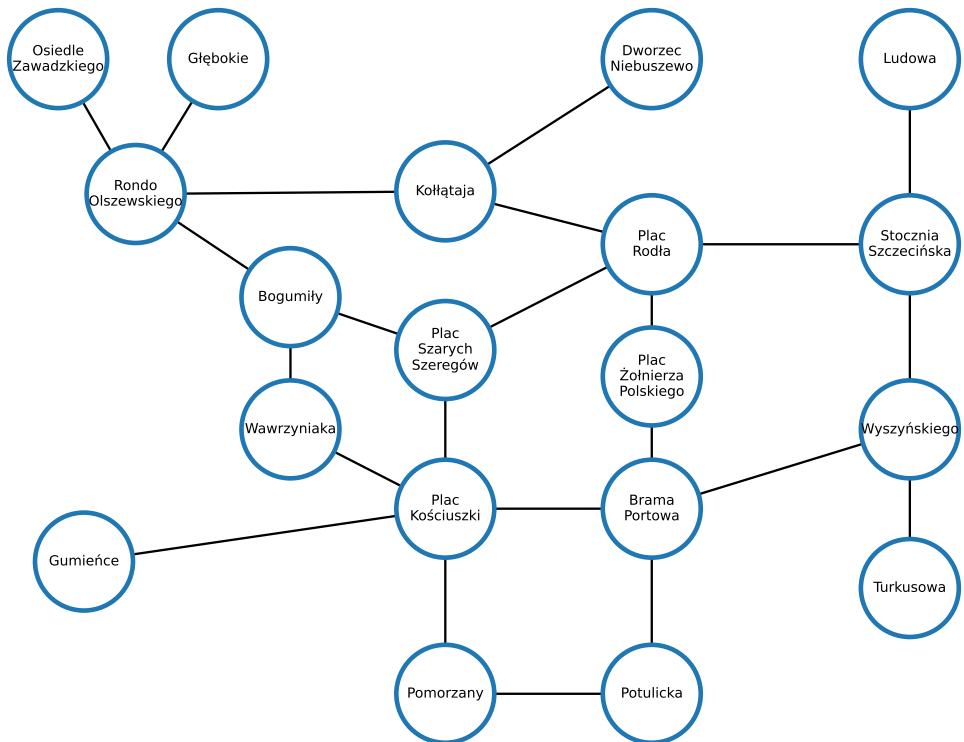
```
1 {
2     'A': {'B': 5, 'C': 3, 'D': 2},
3     'B': {'A': 5, 'D': 3},
4     'C': {'A': 3, 'E': 4},
5     'D': {'A': 2, 'B': 3, 'E': 1},
6     'E': {'C': 4, 'D': 1},
7 }
```

4. Zaimplementować algorytm Dijkstry, służący do wyszukiwania najkrótszej ścieżki w grafie ważonym za pomocą przeszukiwania grafu wrzesz. Przetestować działanie zaimplementowanego algorytmu na dwóch grafach dostarczonych razem z instrukcją (pliki `small_weighted.csv` i `city_weighted.csv`).

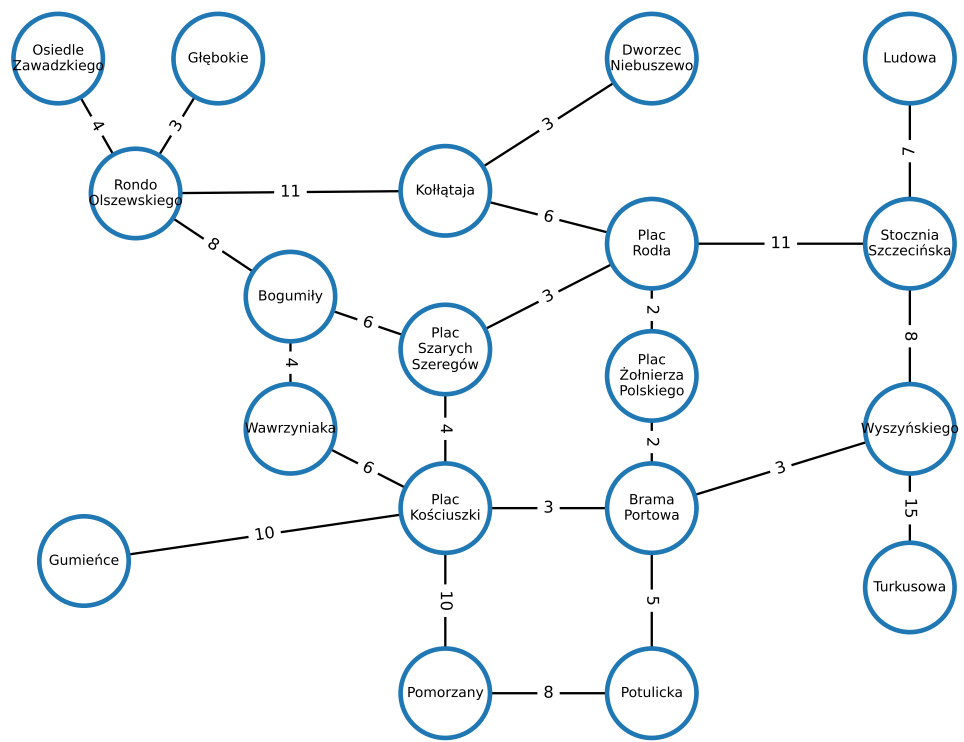
## A Podglądowe wizualizacje grafów z plików



Rysunek 1: Grafy z plików `small.csv` (po lewej) i `small_weighted.csv` (po prawej)



Rysunek 2: Graf z pliku city.csv



Rysunek 3: Graf z pliku city\_weighted.csv