Algorytm Dijkstry

Złożoność obliczeniowa:

Naiwna implementacja algorytmu Dijkstra ma złożoność obliczeniową $O(|V|^{2)}$, natomiast zastosowanie kolejki priorytetowej powoduje że złożoność obliczeniowa maleje do $O(|E|+|V|\log|V|)$

Algorytm Dijkstry:

```
1  function Dijkstra(Graph, start):
2    dist[start] ← 0              // Inicjalizacja
3
4    Q ← PriorityQueue()           // Inicjalizuj kolejkę priorytetową
5
6    for each vertex v in Graph:
7      if v ≠ source
8        dist[v] ← INFINITY        // Nieznana odległość od startu do v
9        prev[v] ← UNDEFINED       // Poprzednik v
11     Q.add(v, dist[v])
12
13
14     while Q is not empty:       // Główna pętla
15         u ← Q.extract_min()     // Usuń I zwróć najbliższy wierzch.
16         for each neighbor v of u:  // Dla sąsiadów u
17             alt = dist[u] + length(u, v)
18             if alt < dist[v]
19                 dist[v] ← alt
20                 prev[v] ← u
21                 Q.updatePriority(v, alt)
22
23     return dist[], prev[]
```

Uwagi do Matlaba:

Ponieważ w Matlabie brakuje kolejki priorytetowej, dlatego w materiałach od prowadzącego dostarczona jest odpowiednia klasa implementująca kolejkę priorytetową PriorityQueue.

Uwaga jest to bardzo naiwna implementacja kolejki priorytetowej dlatego

Przykłady użycia:

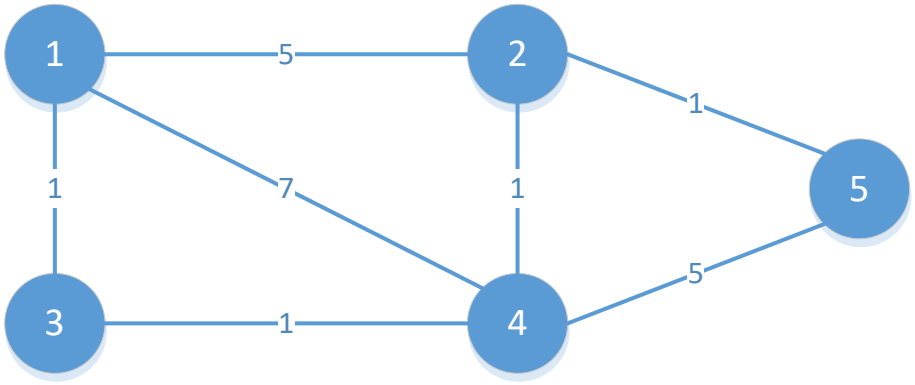
```
q = PriorityQueue();
q = add(q,3,0.3); %Dodanie do kolejki elementu 3 o
wadze 0.3
q = add(q,1,0.1); %Dodanie do kolejki element 1 o
wadze 0.1

size(q) %Zbadanie rozmiaru kolejki
[q,v,p] = remove(q);   %Pobranie głowy kolejki
priorytetowej i usunięcie elementu v z kolejki
```

Zwracane wyniki to: q – kolejka, v – element będący głowa kolejki, p – priorytet elementu v

q = updatePriority(q,v,p); %Aktualizacja priorytetu p element v znajdującego się w kolejce
[V,P] = elements(q) – pobranie listy elementów znajdujących się w kolejce wraz z priorytetami

Zadania

1)
Zapisz poniższy graf w postaci macierzy sąsiedztwa



2)
Zaimplementuj algorytm Dijkstry

3)
Znajdź najkrótszą ścieżkę w podanym grafie między wierzchołkami 1 i 5, ile ona wynosi

4)
Załaduj graf znajdujący się w pliku graph39.mat. Znajdź najkrótszą ściekę i podaj ile ona wynosi pomiędzy węzłami 1 i 10 oraz przez jakie węzły przebiega.

5)
Zaproponuj modyfikację algorytmu tak, aby główna pętla programu jak i sam program kończył się po wyznaczeniu najmniejszej odległości pomiędzy zadaną parą wierzchołków w grafie(Obecnie iteracje trwają aż nie zostanie znaleziona pełna lista odległości do wszystkich wierzchołków)