

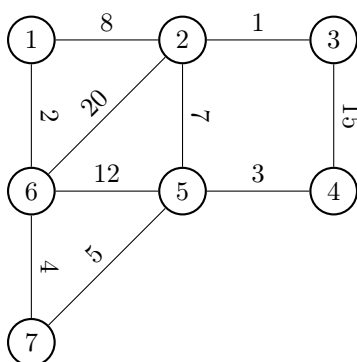
Algorytmy i struktury danych

Na podstawie grafu nieskierowanego zadanego jako następująca lista krawędzi:

(1,2):8 (2,3):1 (3,4):15 (2,5):7 (4,5):3 (5,6):12 (1,6):2 (6,7):4 (2,6):20 (5,7):5

Zadanie 1

Wykonaj rysunek grafu.



Zadanie 2

Znajdź macierz sąsiedztwa.

$$\begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 8 & 0 & 0 & 0 & 2 & 0 \\ 2 & 8 & 0 & 1 & 0 & 7 & 20 & 0 \\ 3 & 0 & 1 & 0 & 15 & 0 & 0 & 0 \\ 4 & 0 & 0 & 15 & 0 & 3 & 0 & 0 \\ 5 & 0 & 7 & 0 & 3 & 0 & 12 & 5 \\ 6 & 2 & 20 & 0 & 0 & 12 & 0 & 4 \\ 7 & 0 & 0 & 0 & 0 & 5 & 4 & 0 \end{pmatrix}$$

Zadanie 3

Zapisz tablicę list sąsiedztwa. Wierzchołki na listach sąsiedztwa powinny być są ustawione rosnąco wg numeru wierzchołka. Ta kolejność powinna być stosowana w symulacji algorytmów DFS, BFS i Dijkstry.

Wierzchołek	Lista sąsiedztwa
1	2, 6
2	1, 3, 5, 6
3	2, 4
4	3, 5
5	2, 4, 6, 7
6	1, 2, 5, 7
7	5, 6

Zadanie 4

Zapisz kolejność odwiedzania wierzchołków przez algorytm DFS startujący z wierzchołka 5.

5, 2, 1, 6, 7, 3, 4

Zadanie 5

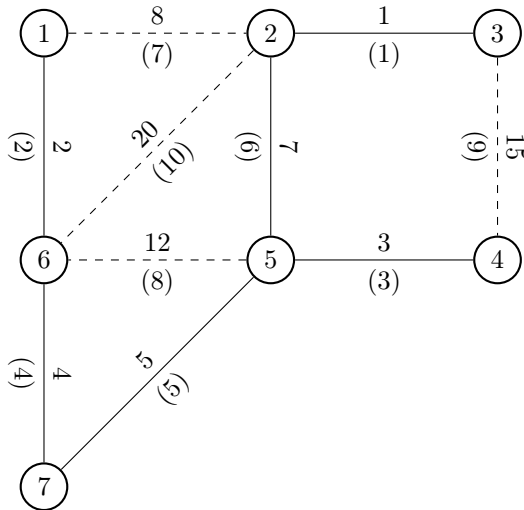
Zapisz kolejność odwiedzania wierzchołków w algorytmie BFS startującym z tego samego wierzchołka.

5, 2, 4, 6, 7, 1, 3

Zadanie 6

(2 pkt) Zasymuluj działanie algorytmu Kruskala i zilustruj rysunkiem:

- liniami przerywanymi oznacz krawędzie nie należące do drzewa wynikowego,
- liniami ciągłymi oznacz krawędzie należące do drzewa wynikowego,
- przy każdej krawędzi w nawiasie okrągłym podaj kolejność w jakiej była ona rozpatrywana.



Zadanie 7

(3 pkt) Zasymuluj działanie algorytmu Dijkstry startując z wierzchołka 3. Zapisz kroki algorytmu podając w każdym kroku:

- numer odwiedzanego wierzchołka
- wykonane w tym kroku operacje `decrease_key` i odpowiednie zmiany w tablicy poprzedników (`prev`)
- wypisując jaka jest zawartość kolejki priorytetowej po wykonaniu kroku

Wierzchołek	Operacje			Kolejka
-	<code>decrease_key(3, 0)</code>	<code>prev[3] = 3</code>		3, 1, 2, 4, 5, 6, 7
3	<code>decrease_key(2, 1)</code> <code>prev[2] = 3</code>	<code>decrease_key(4, 15)</code> <code>prev[4] = 3</code>		2, 4, 1, 5, 6, 7
2	<code>decrease_key(1, 9)</code> <code>prev[1] = 2</code>	<code>decrease_key(5, 8)</code> <code>prev[5] = 2</code>	<code>decrease_key(6, 21)</code> <code>prev[6] = 2</code>	5, 1, 4, 6, 7
5	<code>decrease_key(4, 11)</code> <code>prev[4] = 5</code>	<code>decrease_key(6, 20)</code> <code>prev[6] = 5</code>	<code>decrease_key(7, 13)</code> <code>prev[7] = 5</code>	1, 4, 7, 6
1	<code>decrease_key(6, 11)</code>	<code>prev[6] = 1</code>		4, 6, 7
4				6, 7
6				7
7				-

Na końcu algorytmu dla każdego wierzchołka zapisz:

- odległość od wierzchołka startowego
- numer wierzchołka będącego poprzednikiem

Wierzchołek	1	2	3	4	5	6	7
Odległość	9	1	0	11	8	11	13
Poprzednik	2	3	3	5	2	1	5

Algorytm zilustruj grafem, w którym:

- przy każdym wierzchołku będzie podany w nawiasie okrągłym numer kroku algorytmu, w którym wierzchołek został odwiedzony.
- strzałkami ciągłymi oznaczone będą krawędzie należące do drzewa wynikowego
- strzałkami przerywanymi oznaczone będą krawędzie, które w trakcie algorytmu wskazywały na poprzednika, jednak nie należą do drzewa wynikowego.

