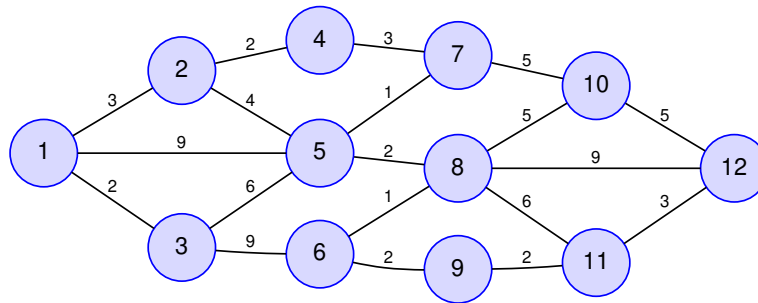


PRZYKŁADOWE WEJŚCIE – ZESTAW 3.

Dane wejściowe: dowolne (tak, żeby cały program w danym uruchomieniu pracował na jednym wygenerowanym losowo grafie z przypisanymi krawędziom wagami).

Ad. 1. Spójny nieskierowany graf losowy z wagami

Wyjście programu: dowolna reprezentacja grafu (przykład: rys. 1).



Rysunek 1: Dane **wyjściowe**: przykładowy losowy graf z wagami z przedziału $[1, 10]$

Ad. 2. Algorytm Dijkstry (zadanie może pobierać numer startowego wierzchołka, ale może też zakładać konkretny numer¹)

Wyjście programu: najkrótsze ścieżki od startowego wierzchołka do pozostałych oraz ich długości (przykład: listing 1).

```
1 START: s = 1
2 d(1) = 0 ==> [1]
3 d(2) = 3 ==> [1 - 2]
4 d(3) = 2 ==> [1 - 3]
5 d(4) = 5 ==> [1 - 2 - 4]
6 d(5) = 7 ==> [1 - 2 - 5]
7 d(6) = 10 ==> [1 - 2 - 5 - 8 - 6]
8 d(7) = 8 ==> [1 - 2 - 4 - 7]
9 d(8) = 9 ==> [1 - 2 - 5 - 8]
10 d(9) = 12 ==> [1 - 2 - 5 - 8 - 6 - 9]
11 d(10) = 13 ==> [1 - 2 - 4 - 7 - 10]
12 d(11) = 14 ==> [1 - 2 - 5 - 8 - 6 - 9 - 11]
13 d(12) = 17 ==> [1 - 2 - 5 - 8 - 6 - 9 - 11 - 12]
```

Listing 1: Dane **wyjściowe**: najkrótsze ścieżki dla grafu z rys. 1. Wierzchołek startowy: $s = 1$; $d(i)$ – długość najkrótszej ścieżki od wierzchołka s do wierzchołka i . Obok długości wypisano ścieżkę w formie numerów poszczególnych wierzchołków, które ją tworzą

Ad. 3. Macierz odległości

Wyjście programu: macierz odległości, np. listing 2.

¹...ponieważ i tak w następnym zadaniu trzeba będzie wywołać algorytm Dijkstry dla pozostałych wierzchołków jako startowych.

1	0	3	2	5	7	10	8	9	12	13	14	17
2	3	0	5	2	4	7	5	6	9	10	11	14
3	2	5	0	7	6	9	7	8	11	12	13	16
4	5	2	7	0	4	7	3	6	9	8	11	13
5	7	4	6	4	0	3	1	2	5	6	7	10
6	10	7	9	7	3	0	4	1	2	6	4	7
7	8	5	7	3	1	4	0	3	6	5	8	10
8	9	6	8	6	2	1	3	0	3	5	5	8
9	12	9	11	9	5	2	6	3	0	8	2	5
10	13	10	12	8	6	6	5	5	8	0	8	5
11	14	11	13	11	7	4	8	5	2	8	0	3
12	17	14	16	13	10	7	10	8	5	5	3	0

Listing 2: Dane **wyjściowe**: macierz odległości dla grafu z rys. 1; element w wierszu w i kolumnie k oznacza długość najkrótszej ścieżki od wierzchołka w do k

Ad. 4. Centrum/centrum minimax

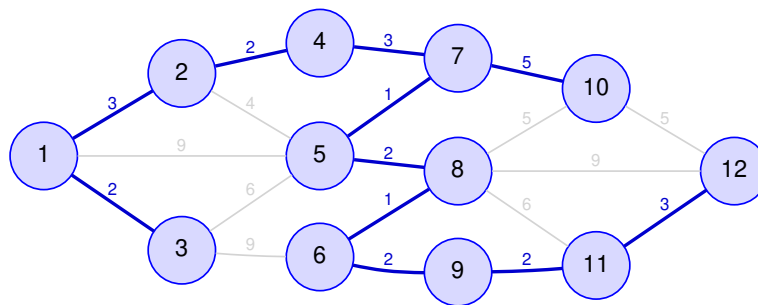
Wyjście programu: numery odpowiednich wierzchołków, np. listing 3.

```
1 Centrum = 5 (suma odleglosci: 55)
2 Centrum minimax = 8 (odleglosc od najdalszego: 9)
```

Listing 3: Dane **wyjściowe**: numery wierzchołków centrum oraz centrum minimax dla grafu z rys. 1

Ad. 5. Minimalne drzewo rozpinające

Wyjście programu: minimalne drzewo rozpinające w dowolnej formie, np. wizualizacji (przykład: rys. 2).



Rysunek 2: Dane **wyjściowe**: minimalne drzewo rozpinające grafu z rys. 1; do drzewa należą tylko pogrubione, niebieskie krawędzie