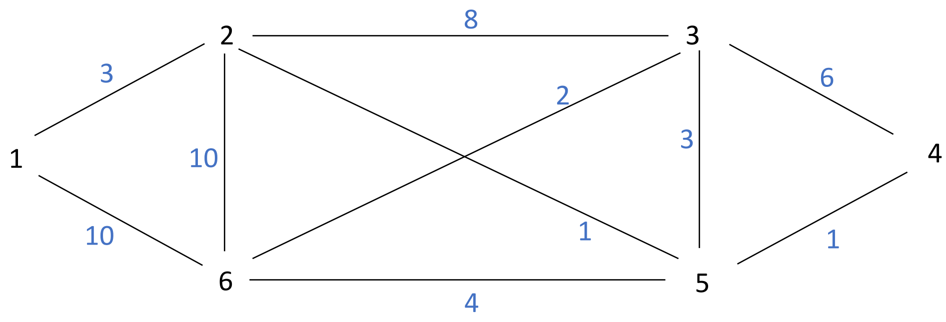
Zadanie 1 - przykład.

Wykorzystamy algorytm Dijkstry, aby znaleźć najkrótsze ścieżki z wierzchołka 1 do wszystkich pozostałych wierzchołków następującego grafu G (wagi krawędzi zaznaczone kolorem niebieskim):



Będziemy używać pomocniczych tablic d i p. Dla wierzchołka v, $d[v]$ będzie długością najkrótszej znanej w danym momencie ścieżki z wierzchołka 1 do v, a $p[v]$ będzie poprzednikiem wierzchołka v na tej ścieżce. Na początku mamy $d[1] = 0$, $p[1] = 0$ ($p[v] = 0$ oznacza brak poprzednika) oraz $d[v] = \infty$ i $p[v] = 0$ dla wszystkich $v \neq 1$. Zatem $d = [0, \infty, \infty, \infty, \infty, \infty]$, $p = [0,0,0,0,0,0]$. Zbiorem wierzchołków, dla których należy znaleźć najkrótsze ścieżki, jest cały zbiór wierzchołków grafu: $B = \{1,2,3,4,5,6\}$. W każdym kroku algorytmu szukamy w zbiorze B wierzchołka k o najniższej wartości $d[k]$ i usuwamy go z B. Następnie dla każdego sąsiada i wierzchołka k znajdującego się w zbiorze B sprawdzamy, czy uda nam się dojść do niego szybciej przez wierzchołek k, czyli czy $d[k] + w(k,i) < d[i]$. Jeśli tak, to aktualizujemy $d[i]$ i ustawiamy $p[i] = k$.

- Krok 1.
  Wierzchołkiem o najniższej wartości $d[k]$ w zbiorze B jest $k = 1$. Mamy:
  $d[1] + w(1,2) = 3 < d[2]$, więc ustawiamy $d[2] = 3, p[2] = 1$;
  $d[1] + w(1,6) = 10 < d[6]$, więc ustawiamy $d[6] = 10, p[6] = 1$.
  Otrzymujemy $B = \{2,3,4,5,6\}$, $d = [0,3, \infty, \infty, \infty, 10]$, $p = [0,1,0,0,0,1]$.
- Krok 2.
  Znajdujemy $k = 2$. Mamy:
  $d[2] + w(2,3) = 11 < d[3]$, więc ustawiamy $d[3] = 11, p[3] = 2$;
  $d[2] + w(2,5) = 4 < d[5]$, więc ustawiamy $d[5] = 4, p[5] = 2$;
  $d[2] + w(2,6) = 13 > d[6]$, więc nie zmieniamy $d[6]$ i $p[6]$.
  Otrzymujemy $B = \{3,4,5,6\}$, $d = [0,3,11, \infty, 4,10]$, $p = [0,1,2,0,2,1]$.
- Krok 3.
  Teraz $k = 5$. Mamy:
  $d[5] + w(5,3) = 7 < d[3]$, więc ustawiamy $d[3] = 7, p[3] = 5$;
  $d[5] + w(5,4) = 5 < d[4]$, więc ustawiamy $d[4] = 5, p[4] = 5$;
  $d[5] + w(5,6) = 8 < d[6]$, więc ustawiamy $d[6] = 8, p[6] = 5$.
  Otrzymujemy $B = \{3,4,6\}$, $d = [0,3,7,5,4,8]$, $p = [0,1,5,5,2,5]$.
- Krok 4.
  Znajdujemy $k = 4$. Mamy:
  $d[4] + w(4,3) = 11 > d[3]$, więc nie zmieniamy $d[3]$ i $p[3]$.
  Otrzymujemy $B = \{3,6\}$, $d = [0,3,7,5,4,8]$, $p = [0,1,5,5,2,5]$.

- Krok 5.
  Znajdujemy $k = 3$. Mamy:
  $d[3] + w(3,6) = 9 > d[6]$, więc nie zmieniamy $d[6]$ i $p[6]$.
  Otrzymujemy $B = \{6\}$, $d = [0,3,7,5,4,8]$, $p = [0,1,5,5,2,5]$.
- Krok 6.
  Teraz $k = 6$, ale w zbiorze B nie ma już żadnych wierzchołków, którym można by poprawić odległości. $B = \{\}$, $d = [0,3,7,5,4,8]$, $p = [0,1,5,5,2,5]$, koniec algorytmu.

Długości znalezionych najkrótszych ścieżek są zapisane w tablicy d. Aby odtworzyć same ścieżki, należy wykorzystać tablicę p. Z danego wierzchołka v cofamy się do jego poprzednika, następnie do jego poprzednika itd., aż dojdziemy do wartości 0 (brak poprzednika). W ten sposób dostajemy wierzchołki ścieżki w odwrotnej kolejności. Na przykład, przy wypisywaniu najkrótszej ścieżki z wierzchołka 1 do 4, dostajemy: $p[4] = 5$, $p[5] = 2$, $p[2] = 1$, $p[1] = 0$. Zatem ścieżka ma postać: $1 \longrightarrow 2 \longrightarrow 5 \longrightarrow 4$.

Wszystkie krawędzie postaci $(v, p[v])$ tworzą w naszym grafie drzewo najkrótszych ścieżek (zaznaczone na poniższym rysunku na czerwono).