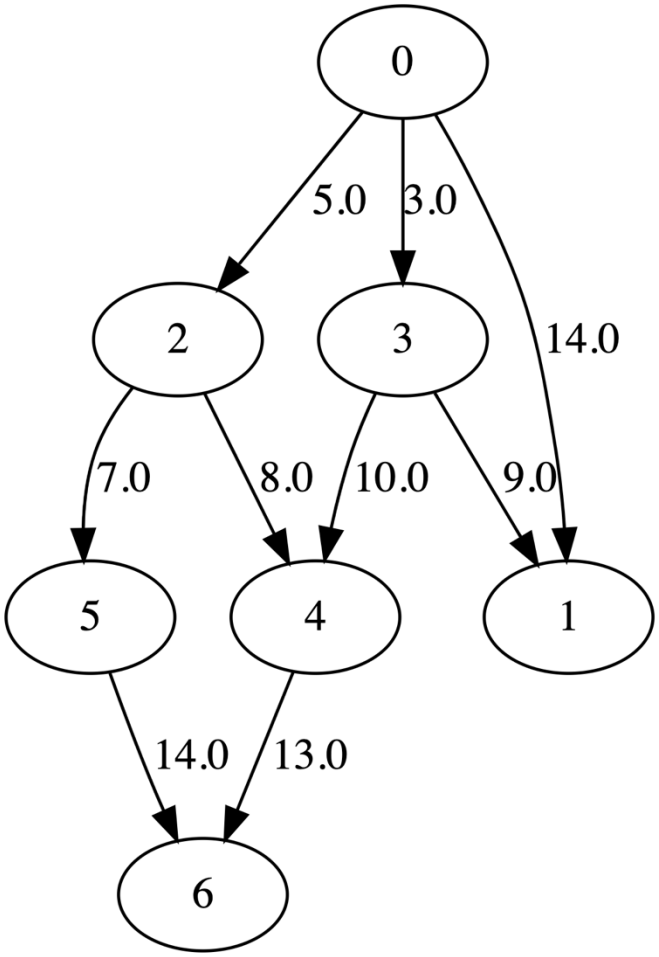
```
Java version used for this program is 12.0.2
GraphTest.java starts

----------7.txt----------
    0    2    3    1    6    4    5
    F    F    F    F    F    F    F
  0.0    L    L    L    L    L    L
    0    2    3    1    6    4    5
Working on Vertex: 0
    0    2    3    1    6    4    5
    T    F    F    F    F    F    F
  0.0  5.0  3.0 14.0    L    L    L
    0    0    0    0    6    4    5
Working on Vertex: 3
    0    2    3    1    6    4    5
    T    F    T    F    F    F    F
  0.0  5.0  3.0  9.0    L 10.0    L
    0    0    0    3    6    3    5
Working on Vertex: 2
    0    2    3    1    6    4    5
    T    T    T    F    F    F    F
  0.0  5.0  3.0  9.0    L  8.0  7.0
    0    0    0    3    6    2    2
Working on Vertex: 5
    0    2    3    1    6    4    5
    T    T    T    F    F    F    T
  0.0  5.0  3.0  9.0 14.0  8.0  7.0
    0    0    0    3    5    2    2
Working on Vertex: 4
    0    2    3    1    6    4    5
    T    T    T    F    F    T    T
  0.0  5.0  3.0  9.0 13.0  8.0  7.0
    0    0    0    3    4    2    2
Working on Vertex: 1
    0    2    3    1    6    4    5
    T    T    T    T    F    T    T
  0.0  5.0  3.0  9.0 13.0  8.0  7.0
    0    0    0    3    4    2    2
Working on Vertex: 6
    0    2    3    1    6    4    5
    T    T    T    T    T    T    T
  0.0  5.0  3.0  9.0 13.0  8.0  7.0
    0    0    0    3    4    2    2
The best way to go from 0 to city 2 is follows
0 -> 2 Cost = 5.0 = 5.0
The best way to go from 0 to city 3 is follows
0 -> 3 Cost = 3.0 = 3.0
The best way to go from 0 to city 1 is follows
0 -> 3 -> 1 Cost = 3.0 + 6.0 = 9.0
The best way to go from 0 to city 6 is follows
0 -> 2 -> 4 -> 6 Cost = 5.0 + 3.0 + 5.0 = 13.0
```



```
The best way to go from 0 to city 4 is follows
0 -> 2 -> 4 Cost = 5.0 + 3.0 = 8.0
The best way to go from 0 to city 5 is follows
0 -> 2 -> 5 Cost = 5.0 + 2.0 = 7.0
WEIGHTED UNDIRECTED GRAPH
Num Vertices = 7
Num Edges    = 12
Work done    = 12
numberofNodeAddedToHeap = 10
Shortest path from city 0 to city 6 = 13.0

----------17.txt----------
    C    D    E    B    G    F    A
    F    F    F    F    F    F    F
    L    L    L    L    L    L  0.0
    C    D    E    B    G    F    A
Working on Vertex: A
    C    D    E    B    G    F    A
    F    F    F    F    F    F    T
  3.0    L    L  1.0    L 10.0  0.0
    A    D    E    A    G    A    A
Working on Vertex: B
    C    D    E    B    G    F    A
    F    F    F    T    F    F    T
  2.0  8.0  6.0  1.0  3.0 10.0  0.0
    B    B    B    A    B    A    A
Working on Vertex: C
    C    D    E    B    G    F    A
    T    F    F    T    F    F    T
  2.0  8.0  5.0  1.0  3.0 10.0  0.0
    B    B    C    A    B    A    A
Working on Vertex: G
    C    D    E    B    G    F    A
    T    F    F    T    T    F    T
  2.0  8.0  5.0  1.0  3.0 10.0  0.0
    B    B    C    A    B    A    A
Working on Vertex: E
    C    D    E    B    G    F    A
    T    F    T    T    T    F    T
  2.0  7.0  5.0  1.0  3.0  7.0  0.0
    B    E    C    A    B    E    A
Working on Vertex: D
    C    D    E    B    G    F    A
    T    T    T    T    T    F    T
  2.0  7.0  5.0  1.0  3.0  7.0  0.0
    B    E    C    A    B    E    A
Working on Vertex: F
    C    D    E    B    G    F    A
    T    T    T    T    T    T    T
  2.0  7.0  5.0  1.0  3.0  7.0  0.0
    B    E    C    A    B    E    A
```

```
The best way to go from A to city C is follows
A -> B -> C Cost = 1.0 + 1.0 = 2.0
The best way to go from A to city D is follows
A -> B -> C -> E -> D Cost = 1.0 + 1.0 + 3.0 + 2.0 = 7.0
The best way to go from A to city E is follows
A -> B -> C -> E Cost = 1.0 + 1.0 + 3.0 = 5.0
The best way to go from A to city B is follows
A -> B Cost = 1.0 = 1.0
The best way to go from A to city G is follows
A -> B -> G Cost = 1.0 + 2.0 = 3.0
The best way to go from A to city F is follows
A -> B -> C -> E -> F Cost = 1.0 + 1.0 + 3.0 + 2.0 = 7.0
WEIGHTED UNDIRECTED GRAPH
Num Vertices = 7
Num Edges    = 26
Work done    = 26
numberofNodeAddedToHeap = 11
Shortest path from city A to city F = 7.0
GraphTest.java Ends
goggle: grapviz online
```