Erik Stinnett

Project III Part III, Erik Stinnett

A. *Write an application that applies Dijkstra's algorithm to produce the shortest path tree for a weighted graph with a given starting node. Test and verify your program with the given graph starting with node A:*

```
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '62998' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 22:
A -> B
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63006' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 9:
A -> C
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63013' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 12:
A -> D
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63018' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 45:
A -> D -> E
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63024' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 51:
A -> C -> F
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63032' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 63:
A -> D -> I -> G
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63039' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 56:
A -> B -> H
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
launcher' '63047' '--' 'c:\Users\eriks\Desktop\A
The best path with a value of 42:
A -> D -> I
PS C:\Users\eriks\Desktop\Algorithms\Project 3>
```

B. *Write a program that produces a minimum spanning tree for a connected weighted graph. Test your program with the given graph above*

```
Minimum Spanning Tree:
Path: 2 - 3 ||Weight: 4
Path: 0 - 2 ||Weight: 9
Path: 4 - 5 ||Weight: 18
Path: 7 - 8 ||Weight: 19
Path: 6 - 8 ||Weight: 21
Path: 0 - 1 ||Weight: 22
Path: 4 - 6 ||Weight: 23
Path: 3 - 8 ||Weight: 30
```

C. *Are a shortest path tree and a minimum spanning tree usually the same?*

No, a shortest path and a minimum spanning tree are not always the same. Minimum Spanning Trees are a spanning tree with the minimum weights of the edges. These trees are not the same as shortest paths, they just denote the smallest weights of the graph of
(Vertex - 1).

Finding the Minimum Spanning Tree:

1. Sort all edge weights in the graph in non-decreasing order
2. Start with the edge weight with the lowest value
3. IF the edge in question does *not* form a cycle, add it to the minimum spanning tree
4. Continue until you've reached V - 1 edges in the minimum spanning tree.
5. Add the total weights together. This is the MST's weight.

As you can see from the above description, creating a MST is necessarily related to the shortest path, as the weights might not be in succession on the graph.

D. *If the graph has an edge with a negative weight, can you apply Dijkstra's algorithm to find a shortest path tree?*

No, Dijkstra's algorithm cannot find a shortest path with negative weights. It will result in an incorrect outcome.