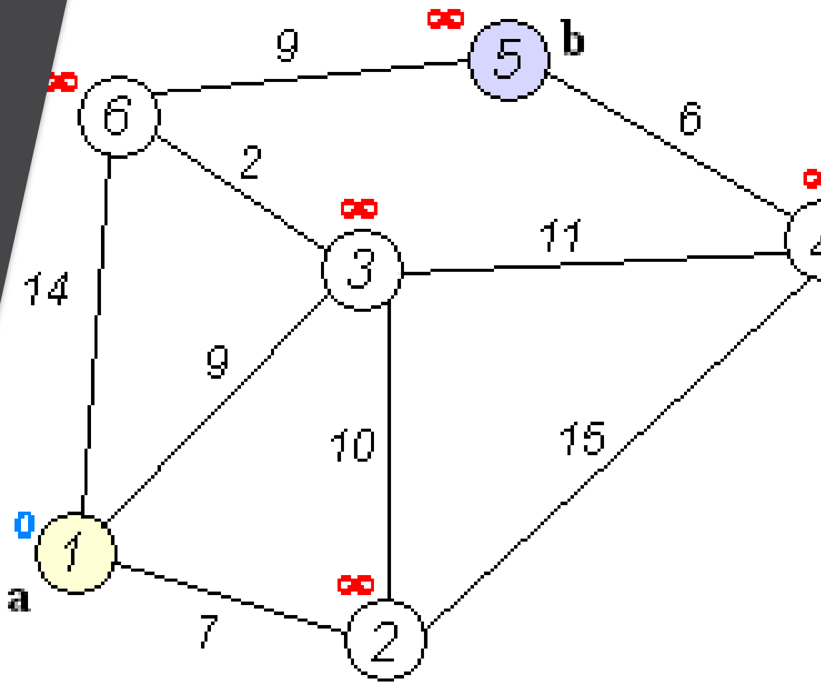
SHORTEST PATH

DIJIKSTRA'S

MAY, 2018



Eric Mercado

Sr Software Engineer

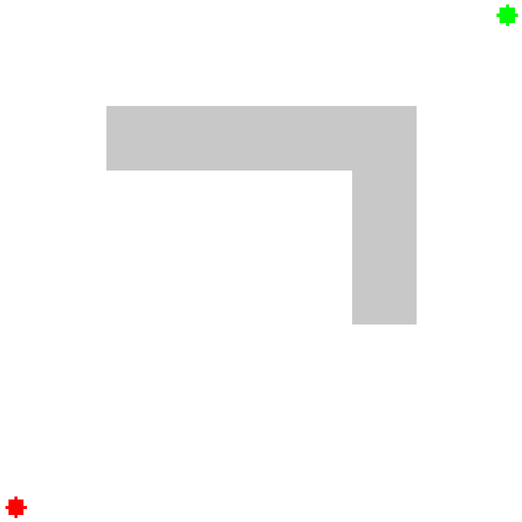
Contents

Overview

Algorithm

Running time

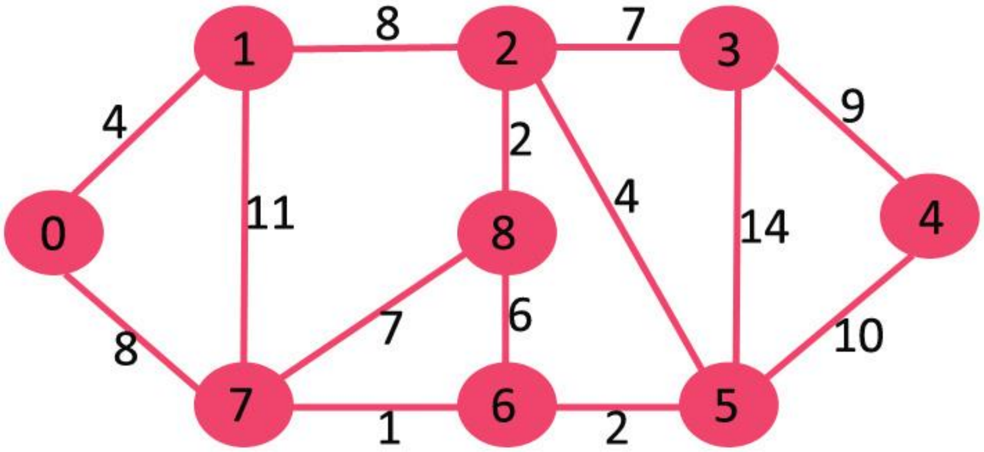
Exercise repo



Overview

- Create a set *sptSet* (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.
- 2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.
- 3) While *sptSet* doesn't include all vertices
  ….**a)** Pick a vertex u which is not there in *sptSet*and has minimum distance value.
  ….**b)** Include u to *sptSet*.
  ….**c)** Update distance value of all adjacent vertices of u.
- To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge

Algorithm Description

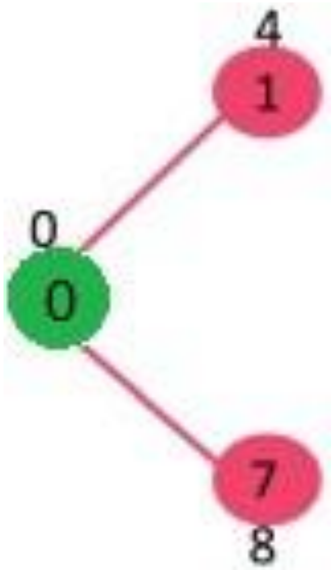


1.-Initial State of the set is {0, INF, INF, INF, INF, INF, INF

Algorith Description

2.- The vertex 0 is picked, include it in *sptSet*. {0}.

update distance values of its adjacent vertices 1 and 7.
The distance values of 1 and 7 are updated as 4 and 8.
Following subgraph shows vertices and their distance values,
The vertices included in SPT are shown in green color.



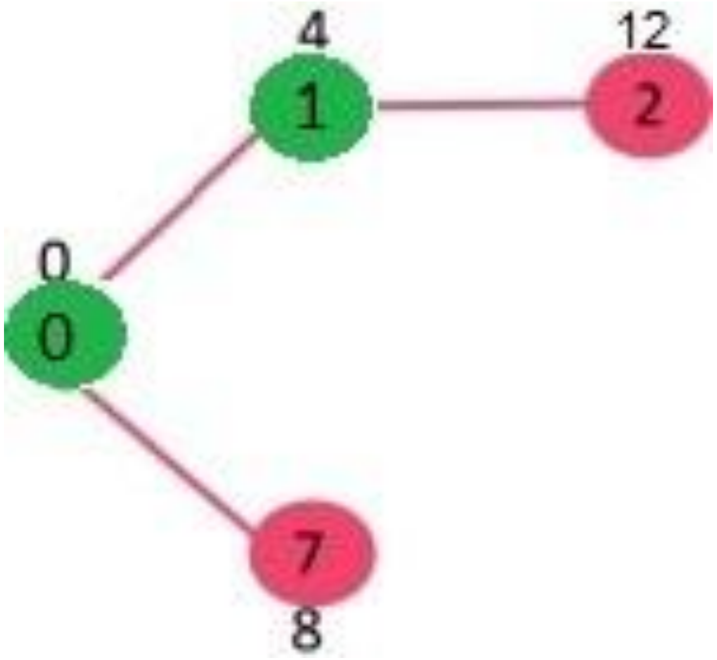
Algorith Description

3.- Pick the vertex with minimum distance value and not already included in SPT.

The vertex 1 is picked and added to sptSet.

So sptSet now becomes {0, 1}. Update the distance values of adjacent vertices of 1

The distance value of vertex 2 becomes 12.



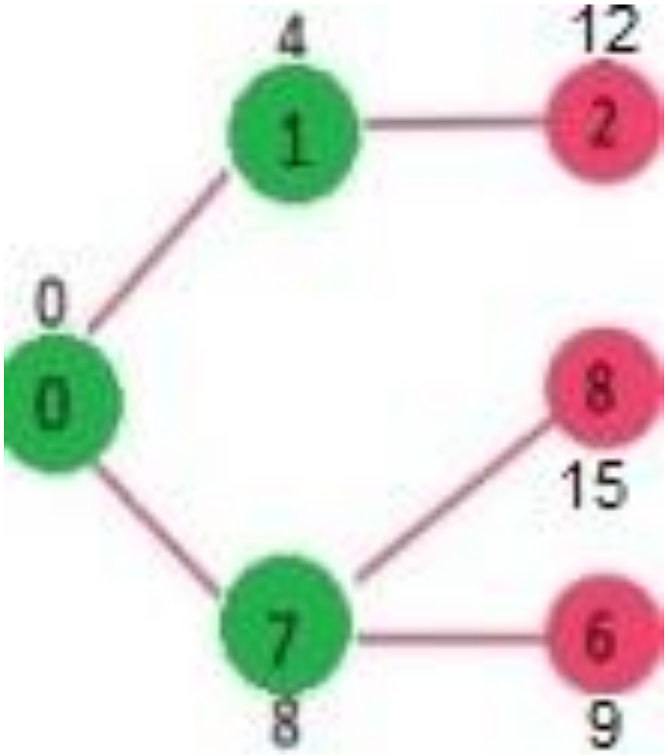
Algorith Description

4.-Pick the vertex with minimum distance value and not already included in SPT.

Vertex 7 is picked. So sptSet now becomes {0, 1, 7}

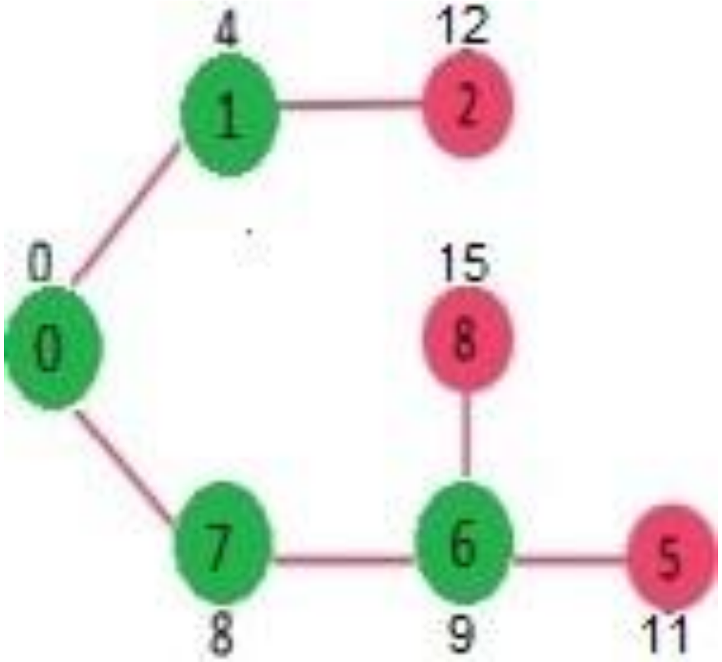
Update the distance values of adjacent vertices of 7.

The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



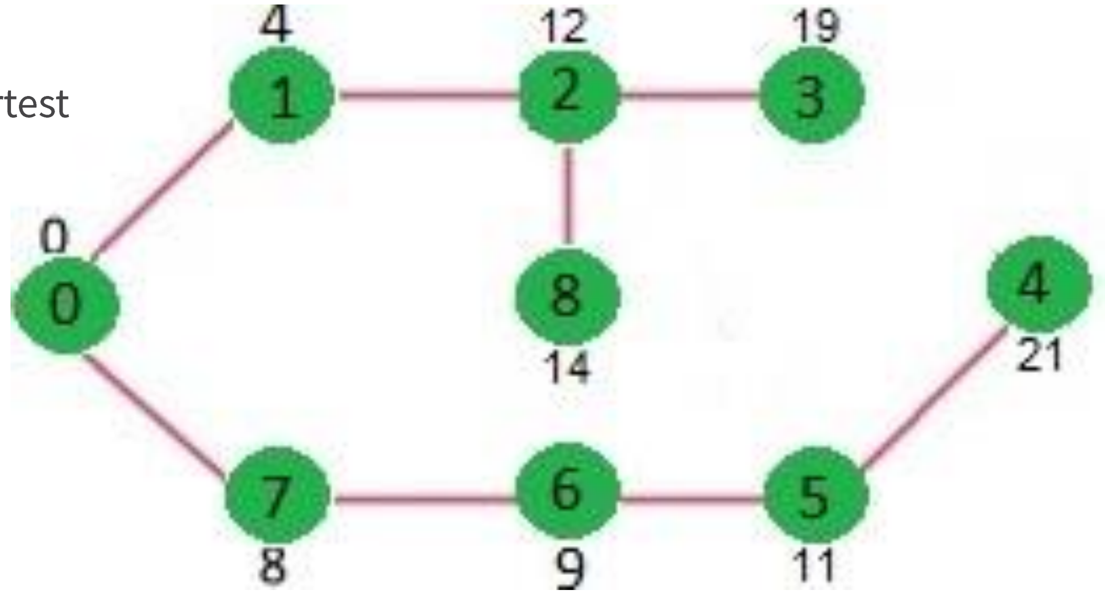
Algorith Description

- 5.- Pick the vertex with minimum distance value and not already included in SPT.
- Vertex 6 is picked. So sptSet now becomes {0, 1, 7, 6}.
- Update the distance values of adjacent vertices of 6.
- The distance value of vertex 5 and 8 are updated



Algorith Description

- We repeat the above steps until *sptSet* doesn’t include all vertices of given graph.
- Finally, we get the following Shortest Path Tree (SPT).



Running time

can be expressed as a function of the number of

edges, **|E|**,

and

vertices **|V|,**

using big-O notation. How tight a bound is possible depends on the way the vertex set Q is implemented. In the following, upper bounds can be simplified because **|E|=O(|V|^2)** for any graph,

but that simplification disregards the fact that in some problems, other upper bounds on |E| may hold.

For any implementation of the vertex set Q, the running time is in

$$O(|E| * T_{dk} + |V| * T_{em})$$

Code Repo

- https://github.com/Eric-mercado/shortestPath/blob/master/src/main/java/ShortestPath.java

THANK YOU!

Eric_mercado@epam.com

May, 2018