

# Problem Set 3

Noella James

02/11/2017

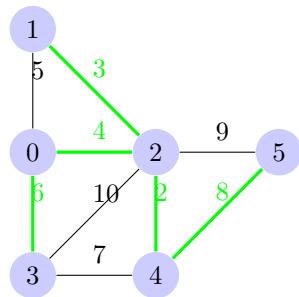
collaborators: none

## 1 Problem 3-2: Shortest Paths and Spanning Trees.

### 1.1 Part a

Prove or disprove. Given an undirected, connected, weighted graph, if you run an MST algorithm on the graph, then the weight of the shortest path between any two vertices is the weight of the unique path between them in the tree.

*Proof.* This statement is false. We will disprove it using a counterexample. Consider the following graph:



The green edges are the edges chosen by the MST. The weight of the shortest path in the graph between vertices 0 and 1 is 5. However, in the MST, the edge (0,1) is not present. Therefore, the shortest path between 0 and 1 is through vertex 2. Therefore, the total weight is 7. Therefore, this disproves the claim.  $\square$

## 1.2 Part b

Prove or disprove. Let  $G = (V, E)$  be an undirected, connected, weighted graph. Suppose we run Dijkstra's algorithm on  $G$  starting at a vertex  $u \in V$ . For every vertex  $v \in V$  let  $p_v$  denote the edges of the shortest path from  $u$  to  $v$  that is found by Dijkstra. Let  $E' = \bigcup_{v \in V} p_v$ .

### 1.2.1 $(V, E')$ is a spanning tree of $G$

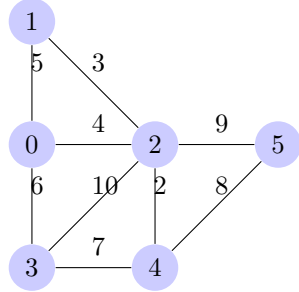
*Proof.* In a Spanning Tree, it is guaranteed that for a vertex  $s$ , there will always be a path to a different vertex  $t$ . Running Dijkstra's algorithm at the start node  $u$  on the undirected Graph  $G$ , it is guaranteed that for every vertex  $v \in V$ , there is a path from vertex  $u$  to  $v$ . Since it is undirected, there is also a path for every  $v \in V$  to  $u$ .

Therefore, if  $s$  and  $t$  do not share an edge  $(s, t)$ , there is still a path from vertex  $s$  to vertex  $t$  through the start node  $u$ .  $\square$

### 1.2.2 $(V, E')$ is a minimum spanning tree of $G$

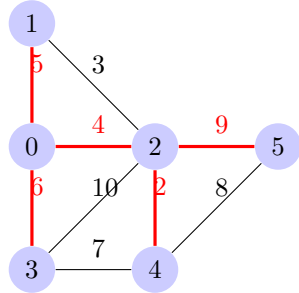
*Proof.* We disprove this claim using a proof by counterexample.

Original Graph:



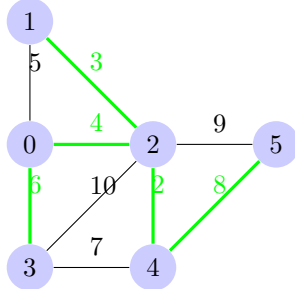
Dijkstra of Graph with start node 0:

The edges in red are the shortest paths to all vertices from vertex 0.



MST of Graph:

The edges in green are the edges in the MST.



The total sum of weights using Dijkstra's algorithm is 26. The total sum of weights using MST is 23. Therefore, since the sum of the weights of Dijkstra's algorithm is more than that of the MST, we can prove that Dijkstra's algorithm does not always provide an MST.

□