Data Structures and Algorithms – (COMP SCI 2C03)
Winter 2021
Tutorial - 7

March 22, 2021

1. How can the number of strongly connected components of a graph change if a new edge is added?
   Answer: Adding an edge either keep the number of connected components unchanged if it is intra-component edge (between the nodes of one component), or decrease it if it is an inter-component edge (connecting a node in one component to a node in another component).

2. Compute the strongly connected components of the digraph G given in Figure 1 using the Kosaraju-Sharir algorithm. In particular, first compute the reverse postorder for G^R. Then run DFS on the reverse postorder obtained from the previous step to compute all the connected components of G.

   Answer: Figure 2 represents G^R (the reverse of the graph shown in Figure 1).
   The post order of G^R starting from node 0: 7-2-1-9-8-6-0-4-5-3
   The reverse post order of G^R: 3-5-4-0-6-8-9-1-2-7
   Labeled nodes by DFS on G from the source 3: 3-4-5, next unlabled node in above list is 0
   Labeled nodes by DFS on G from the source 0: 0-7-2-1, next unlabled node in above list is 6
   Labeled nodes by DFS on G from the source 6: 6-9-8

   So there are three strong components, including "3-4-5", "0-7-2-1", and "6-9-8"

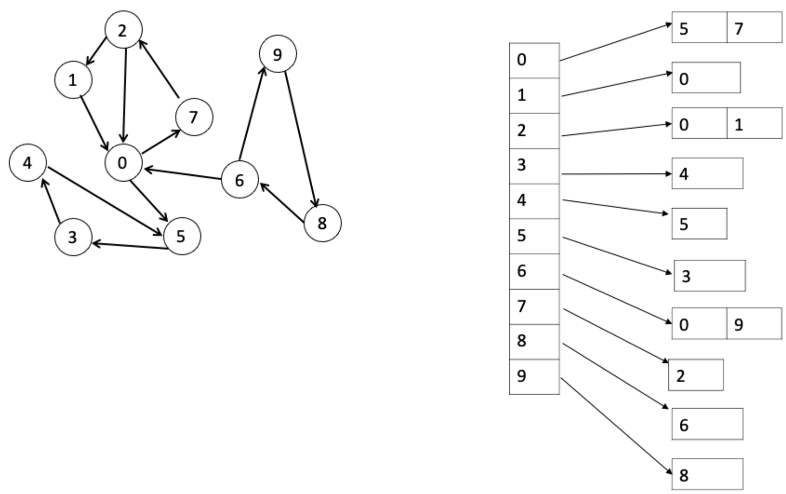

Figure 1: Digraph and its adjacency list for Question 2

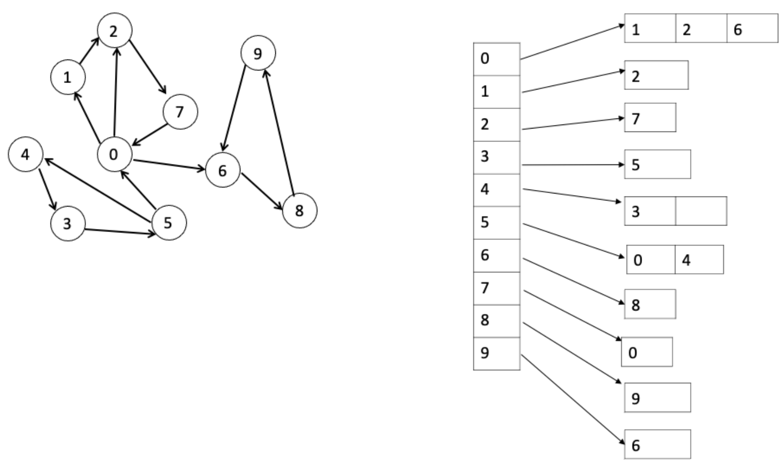

Figure 2: Reverse Digraph G^R and its adjacency list

3. If we modify the Kosaraju-Sharir algorithm to run the first depth-first search in the digraph G (instead of the reverse digraph G^R) and the second depth-first search in G^R (instead of G), then it will still find the strong components.
   Answer: True, the strong components of a digraph are the same as

the strong components of its reverse.

4. Compute the MST of the undirected edge-weighted graph shown in the Figure 3 using

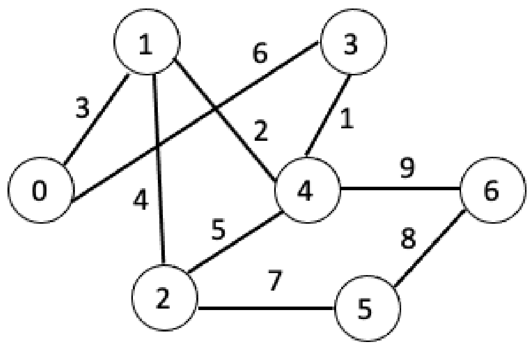

Figure 3: Undirected weighted edge graph

   i. Kruskal's Algorithm
   Answer: We first order the egdes in ascending order.
   3-4 1
   1-4 2
   0-1 3
   1-2 4
   2-4 5
   0-3 6
   2-5 7
   5-6 8
   4-6 9
   Add edges 3-4, 1-4, 0-1, 1-2 to the MST T. Since the edges 2-4, 0-3 create the cycles 1-2-4-1, 0-3-4-1-0, we disregard them. Next we add edges 2-5, 5-6 to the MST. The next edge 4-6 creates the clycle 4-6-5-2-1-4, we dont add it to T.

   ii. Prim's Algorithm
   Answer: We consider edges in the following order:

0-1 3
0-3 6
1-2 4
1-4 2
2-4 5
2-5 7
3-4 1
4-6 9
5-6 8

Let S be the set of vertices so far included in the MST during the algorithms execution. We begin with the vertex 0, that is, with $S = \{0\}$. Consider edges 0-1, 0-3 incident on 0. We add 0-1 to T as it has the smallest weight. Now $S = \{0, 1\}$. We consider the below edges incident on 0 and 1 (disregarding edges already in the MST).
0-3 6
1-2 4
1-4 2

We add edge 1-4 to T as it has min. weight in the above list. Now $S = \{0, 1, 4\}$. We consider the below edges incident on 0, 1 and 4 (disregarding edges already in the MST).
0-3 6
1-2 4
2-4 5
3-4 1
4-6 9

We add edge 3-4 to T as it has min. weight in the above list. Now $S = \{0, 1, 3, 4\}$. We consider the below edges incident on 0, 1, 3 and 4(disregarding edges already in the MST, and edges connecting points in S, for example the edge 0-3).
1-2 4
2-4 5
4-6 9
We add edge 1-2 to T as it has min. weight in the above list.

Now $S = \{0, 1, 2, 3, 4\}$. We consider the below edges incident on 0, 1, 2, 3 and 4(disregarding edges already in the MST, and edges connecting points in S, for example the edge 2-4).
2-5 7
4-6 9
We add edge 2-5 to T as it has min. weight in the above list. Now $S = \{0, 1, 2, 3, 4, 5\}$. We consider the below edges incident on 0, 1, 2, 3, 4 and 5 (disregarding edges already in the MST).
4-6 9
5-6 8
We add edge 5-6 to T as it has min. weight in the above list. Now $S = \{0, 1, 2, 3, 4, 5, 6\}$, and since we have V-1=6 edges added. We stop here. The resulting MST in this step is the MST for G.

5. How would you find a maximum spanning tree of an edge-weighted graph?
Answer: The maximum spanning tree of an edge-weighted connected graph can be computed using the Khruskal's algorithm, by arranging the weight in decending order (instead of ascending), and adding the largest edge each time, unless it creates a cycle till V-1 edges are added to the MST T.

6. Consider the assertion that an edge-weighted graph G has a unique MST only if its edge weights are distinct. Give a proof or a counterexample.
Answer: Let P = an edge-weighted graph G has a unique MST, and let Q = G's edge weights are distinct. Then the statement in the question is of the form P only if Q, which is equivalent to $P \rightarrow Q$, and which is also equivalent to $\neg Q \rightarrow \neg P$. The last implication is easy to show, and it states that "If G's egdes weights are NOT distinct then G (an edge weight graph) does not have a unique MST". This statement is not true. Figure 4 provides the counter example. In the figure, we have two edges (AB and CD) with equal weights. However, the MST consisting of edges A-B, B-C, and C-D is the unique MST having weight 4.

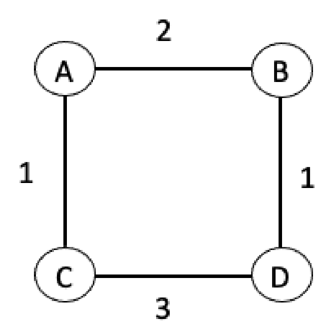

Figure 4: Counter example for the assertion in Q6