Bryan O’Hagan

April 13, 2015

CSCI 3104 Algorithms

TA: William Mortl

Assignment #8

**Problem 1:**

(A)

(B)

|  |  |
| --- | --- |
| Start Node | Nodes visited |
| n1 | {n1} |
| n2 | {n1,n2} |
| n3 | {n6,n5,n4,n2,n1,n3} |
| n4 | {n2, n1, n4 ,n5, n6} |
| n5 | {n4,n2,n1,n5} |
| n6 | {n6,n4,n2,n1,n5} |
| n7 | {n2,n1,n4,n6,n5,n7} |
| n8 | {n1,...,n11} |
| n9 | {n1,...,n11} |
| n10 | {n1,...,n11} |
| n11 | {n1,...,n11} |

**Problem 2:**

Since this network of students is represented in strongly connected undirected graph, every student is able to contact every other student. Just like in depth-first search I would mark every student that has seen the message as a gray node. As the professor, I would require every student to send the message to another student that has not seen the message (those that have not seen the message are white nodes). If a student has sent the message to two other students (unless everyone has seen the message) and seen the message themselves then that student will be a black node. Once every student node is black, then every student has seen the message once.

**Problem 3:**

(A)

(B)

These are the sequence of edges added in order by Kruskal’s algorithm:

MST: { (n1,n4), (n4,n5),(n4,n6),(n6,n3), (n8,n9), (n4,n2), (n8,n10), (n3,n8), (n10,n11), (n2,n7)}

These edges were considered but not added: { (n1,n2), (n5,n6), (n9,n11) }

**Problem 4:**

(A) From whatever node we start from in a weighted graph, prim’s algorithm looks at all the nodes connecting to our starting node and chooses the edge with the smallest weight. The algorithm adds these two nodes and edge between the two to our final spanning tree then looks for the smallest edge again connecting to the new two node spanning tree. This recursively happens until every single node is connected. Since e: (s,t) has the smallest weight, it will always in our final spanning tree no matter what position it’s in.

(B) Kruskal’s algorithm looks at every node in the graph and connects nodes with the smallest weight between the two. It then connects nodes with the next smallest weight. All these forests of nodes will come together to produce a minimum spanning tree. Since e: (s,t) has the smallest weight then it will be chosen first to be in the final spanning tree.