COMPSCI/SFWRENG 2C03

Data Structures and Algorithms

Ryszard Janicki

McMaster University

Assignment 3

Name: Hishmat Salehi

MacId: Salehh6

Student number: 400172262

1. a. 0: $5 \to 2 \to 6$
   1: $4 \to 8 \to 11$
   2: $5 \to 6 \to 0 \to 3$
   3: $10 \to 6 \to 2$
   4: $1 \to 8$
   5: $0 \to 10 \to 2$
   6: $2 \to 3 \to 0$
   7: $8 \to 11$
   8: $1 \to 11 \to 7 \to 4$
   9:
   10: $5 \to 3$
   11: $8 \to 7 \to 1$

   b. Adjacency Matrix:

```
0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
```

2.

a)

dfs(0)

dfs(5)

check 0

dfs(10)

check 5

dfs(3)

check 10

dfs(6)

dfs(2)

check 5

check 6

check 0

check 3

2 done

check 3

check 0

6 done

check 2

3 done

10 done

check 2

5 done

check 2

check 6

0 done

b)

0

5

10

3

6

2

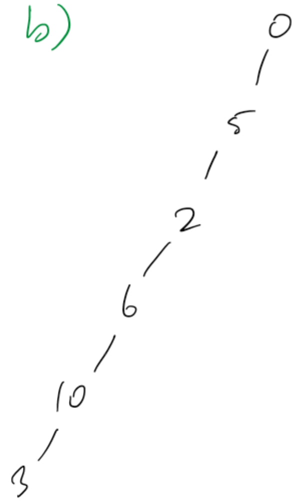
3.

a)

bfs(0)

- bfs(5)
  - check 0
  - bfs(10)
    - check 5
    - check 3
  - check 2
- bfs(2)
  - check 5
  - check 6
  - check 0
  - bfs(3)
    - check 10
    - check 6
    - check 2
- bfs(6)
  - check 2
  - check 3
  - check 0

b)



4. Consider by contradiction that the edge of maximum weight in the cycle C, edge e, belongs to the MST of the graph. Since MSTs do not contain cycles there is at least one edge in C that is not in the MST. Let's call one of these edges f. Now add f to the MST. There is now a cycle in the MST. Since e has the maximum weight in the cycle C and all edge weights are distinct, it means that weight(f) ¡ weight(e). Removing the edge e after having added the edge f would generate a new MST' with total weight less than the total weight in MST, contradicting its minimality.

5. a. 0: $6 \to 5$
   1:
   2: $0 \to 3$
   3: $10 \to 6$
   4: 1
   5: $10 \to 2$
   6: 2
   7: $8 \to 11$
   8: $1 \to 4$
   9:
   10: 3
   11: 8

   b. Adjacency Matrix:

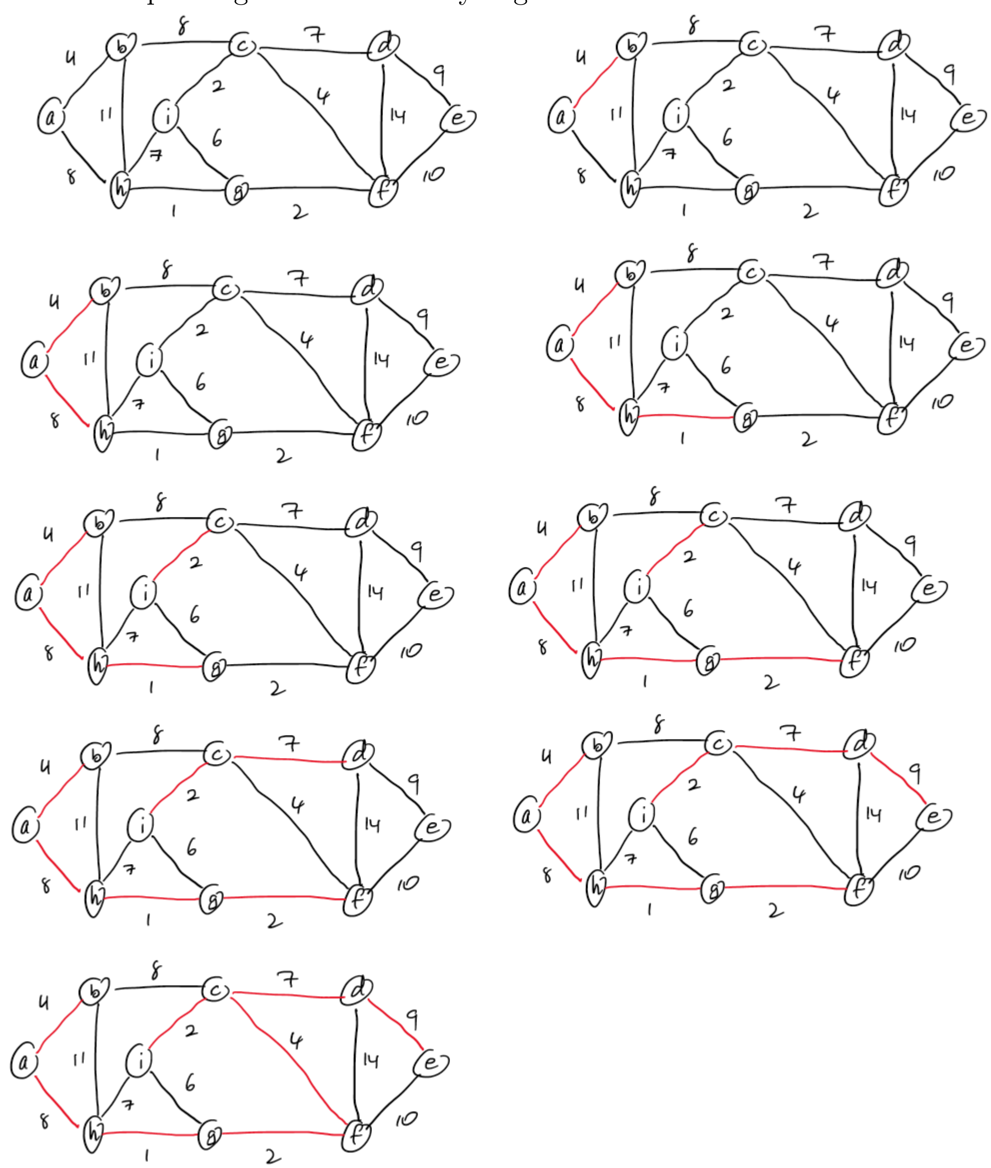
```
0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
```

6. [TODO: Show steps pg. 589] It's strongest component is 0 2 3 5 6 10.

7. Topological order:
   p − n − o − s − m − r − u − y − v − w − z − q − t − x

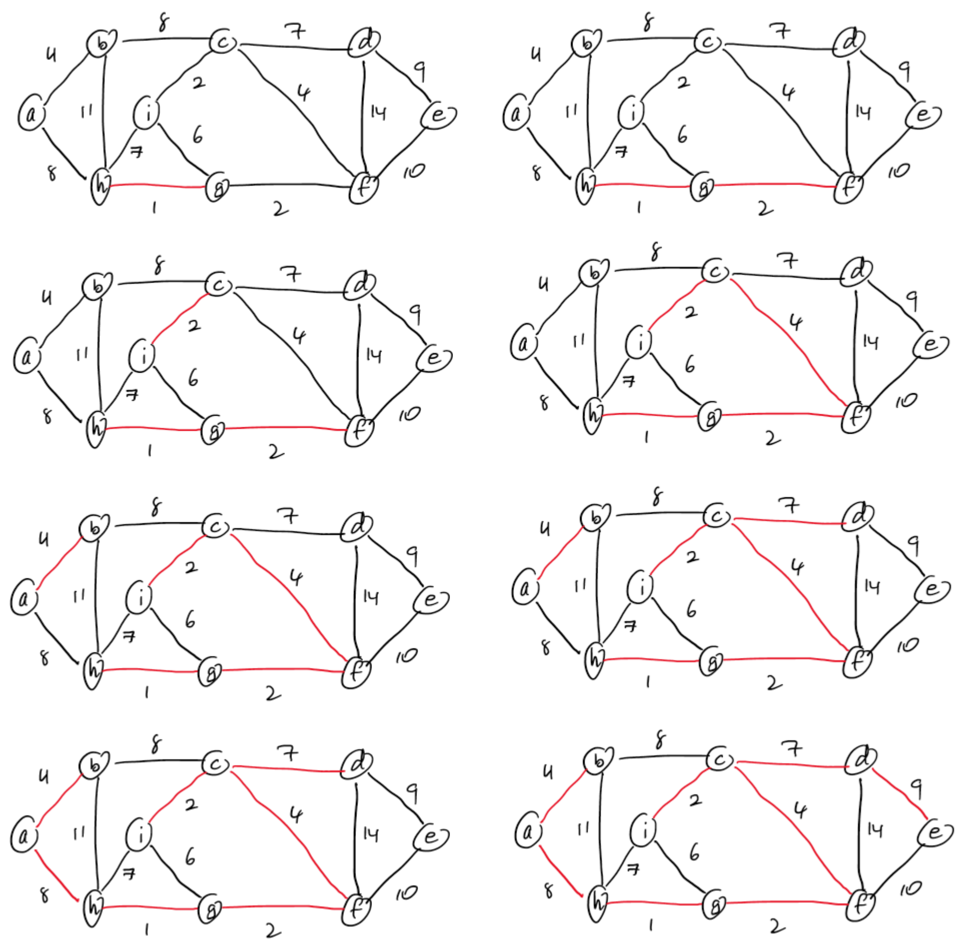
8. Suppose there are two minimum trees, A and B. Let e be the edge in just one of A,B with the smallest cost. Suppose it is in A but not B. Suppose e is the edge PQ. Then B must contain a path from P to Q which is not simply the edge e. So if we add e to B, then we get a cycle. If all the other edges in the cycle were in A, then A would contain a cycle, which it cannot. So the cycle must contain an edge f not in A. Hence, by the definition of e (and the fact that all edge-costs are different) the cost of f must be greater than the cost of e. So if we replace f by e we get a spanning tree with smaller total cost. Contradiction.

9. a. Minimum spanning tree with Greedy Algorithm:



b. Minimum spanning tree with Kruskal's Algorithm:

Monday, April 6, 2020
3:16 AM



c. Minimum spanning tree with Prim's Algorithm: