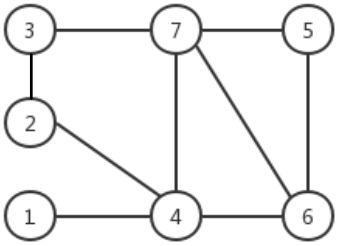
Data Structures and Algorithms
INFO 6205
Homework 9
Due: March 31, 2019

Put all your java, compiled class files and documentation files into a zip file Homework9.zip and submit it via the dropbox on the blackboard before the END of due date. Put your name on all .java files. There will be a short Quiz on this homework.

1. a) Draw the undirected graph that is represented as follows:
   Vertices: 1, 2, 3, 4, 5, 6, 7
   Edges: (5, 6), (4, 6), (3, 7), (6, 7), (5, 7), (1, 4), (2, 4), (2, 3), (4, 7)

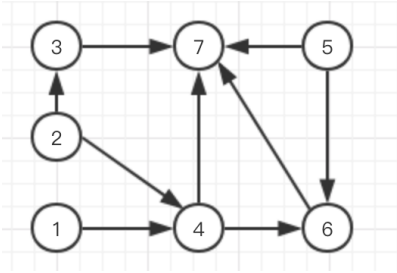


b) Is the graph connected? Is it complete? Explain.

This graph is connected. Because if a graph is connected, then for any two vertices i, j in the graph, they can connect to each other through a certain path. In this diagram, any two vertices can reach each other through a path. So, this graph is connected.

This graph is not complete, because some vertices are not directly connected. Like 3-4, 1-2..etc.

c) Draw the Directed graph using the same data as above.

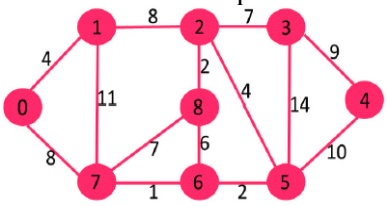


d) Is the directed graph connected? Is it complete? Explain.

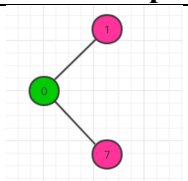
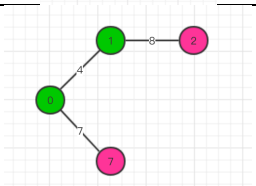
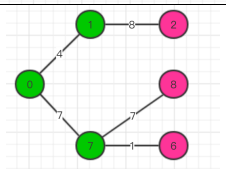
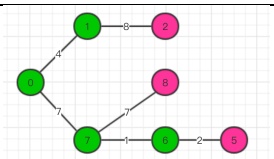
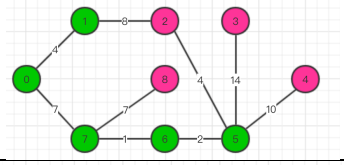
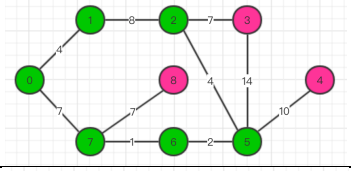
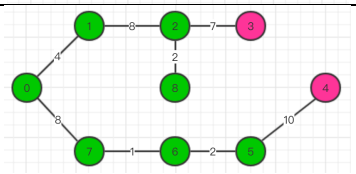
This graph is not connected. Because if a graph is connected, then for any two vertices i, j in the graph, they can connect to each other through a certain path. In this diagram, some two vertices can't reach each other through a path. So, this graph is not connected.

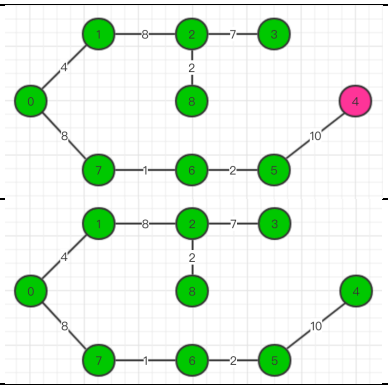
This graph is not complete, because some vertices are not directly connected. Like 3-4, 1-2..etc.

2. Consider this Undirected Graph:



a) What is the shortest-path of this graph, show **step-by-step** algorithm
sptSet: shortest path tree set
INF: Infinite
S: Record the vertex of the shortest path (and the corresponding shortest path length)
U: Record the vertex of the shortest path (and the distance from the vertex to the starting point s)

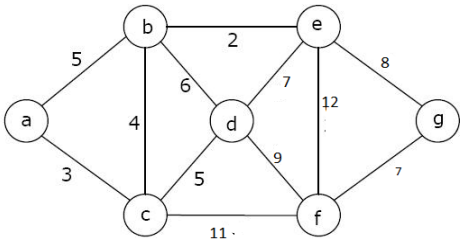
Show Step	Step Detail
	Select the top vertex 0 S = {0(0)} U = {1(4), 2(INF), 3(INF), 4(INF), 5(INF), 6(INF), 7(8), 8(INF)} sptSet = {0}
	Select the next vertex 1 S = {0(0), 1(4)} U = {2(12), 3(INF), 4(INF), 5(INF), 6(INF), 7(8), 8(INF)} sptSet = {0, 1}
	Select the next vertex 7 S = {0(0), 1(4), 7(8)} U = {2(12), 3(INF), 4(INF), 5(INF), 6(9), 8(15)} sptSet = {0, 1, 7}
	Select the next vertex 6 S = {0(0), 1(4), 7(8), 6(9)} U = {2(12), 3(INF), 4(INF), 5(11), 8(15)} sptSet = {0, 1, 7, 6}
	Select the next vertex 5 S = {0(0), 1(4), 7(8), 6(9), 5(11)} U = {2(12), 3(24), 4(21), 8(15)} sptSet = {0, 1, 7, 6, 5}
	Select the next vertex 2 S = {0(0), 1(4), 7(8), 6(9), 5(11), 2(12)} U = {3(19), 4(21), 8(15)} sptSet = {0, 1, 7, 6, 5, 2}
	Select the next vertex 8 S = {0(0), 1(4), 7(8), 6(9), 5(11), 2(12), 8(14)} U = {3(19), 4(21)} sptSet = {0, 1, 7, 6, 5, 2, 8}

	Select the next vertex 3 S = {0(0), 1(4), 7(8), 6(9), 5(11), 2(12), 8(14), 3(19)} U = {4(21)} sptSet = {0, 1, 7, 6, 5, 2, 8, 3}
	Select the next vertex 4 S = {0(0), 1(4), 7(8), 6(9), 5(11), 2(12), 8(14), 3(19), 4(21)} U = {} sptSet = {0, 1, 7, 6, 5, 2, 8, 3, 4}

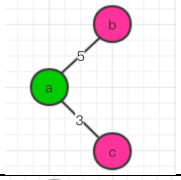
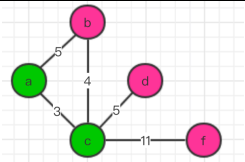
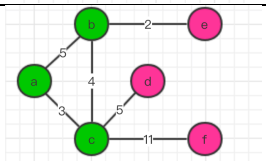
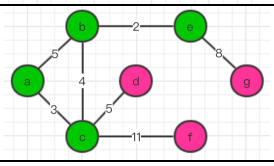
b) What is the space and time complexity of this algorithm?
Time Complexity of the implementation is O(V^2). If the input graph is represented using adjacency list, it can be reduced to O(E log V) .

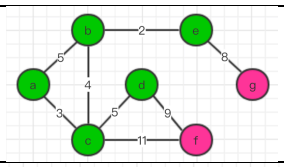
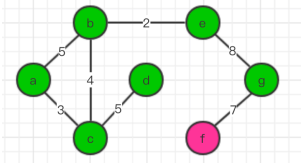
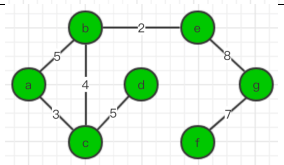
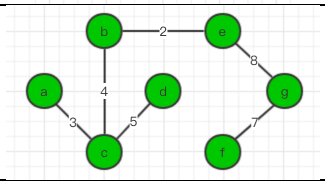
c) Write Java code, compile and test.

3. Solve the Minimum Spanning Tree for the following Graph,



a) Prim's Algorithm step-by-step

Show Step	Step Detail
	Select the top vertex a S = {a(0)} U = {b(5), c(3), d(INF), e(INF), f(INF), g(INF)} mstSet = {a}
	Select the next vertex c S = {a(0), c(3)} U = {b(4), d(5), e(INF), f(11), g(INF)} mstSet = {a, c}
	Select the next vertex b S = {a(0), c(3), b(4)} U = {d(5), e(2), f(11), g(INF)} mstSet = {a, c, b}
	Select the next vertex e S = {a(0), c(3), b(4), e(2)} U = {d(5), f(11), g(8)} mstSet = {a, c, b, e}

	Select the next vertex d S = {a(0), c(3), b(4), e(2), d(5) } U = {f(9), g(8)} mstSet = {a, c, b, e, d}
	Select the next vertex g S = {a(0), c(3), b(4), e(2), d(5) , g(8)} U = {f(7)} mstSet = {a, c, b, e, d, g}
	Select the next vertex g S = {a(0), c(3), b(4), e(2), d(5) , g(8), f(7)} U = {} mstSet = {a, c, b, e, d, g, f}
	mstSet = {a, c, b, e, d, g, f} The order is a - c - b - e - d - g - f

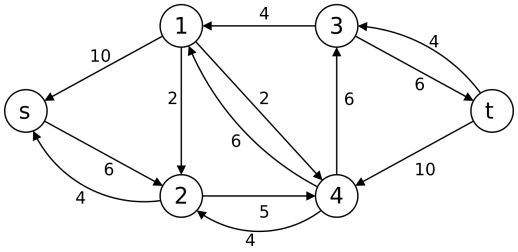
b) Write Java code for algorithm (a)

c) Space and Time complexity?

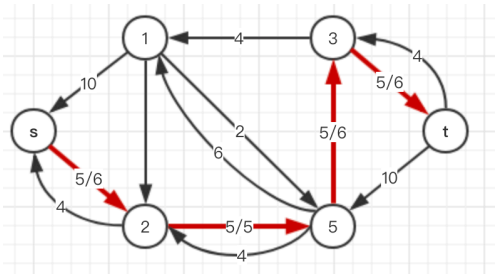
Time Complexity of the above program is O(V^2). If the input graph is represented using adjacency list, then the time complexity of Prim's algorithm can be reduced to O(ElogV).

4. Consider Capacity, Flow, Residual Capacity, and Augmented Path in the following

Ford-Fulkerson Graph:



a) What is the Maximum Flow in the graph, show ALL augmented paths step-by-step



The max flow is 5.

route	capability	flow	
s-2	6	5	0
2-4	5	5	1
4-3	6	5	0
3-t	6	5	1

b) What is the Time complexity and Space of the algorithm?

The time complexity of the above implementation is O(EV3)

c) What is the Edmonds-Karp contribution to Fulkerson method, and Why?

The Edmonds-Karp Algorithm is a specific implementation of the Ford-Fulkerson algorithm. Like Ford-Fulkerson, Edmonds-Karp is also an algorithm that deals with the max-flow min-cut problem.

Ford-Fulkerson is sometimes called a method because some parts of its protocol are left unspecified. Edmonds-Karp, on the other hand, provides a full specification. Most importantly, it specifies that breadth-first search should be used to find the shortest paths during the intermediate stages of the program.

Edmonds-Karp improves the runtime of Ford-Fulkerson, which is O(|E|f*), to O(|V||E|^2). This improvement is important because it makes the runtime of Edmonds-Karp independent of the maximum flow of the network.

d) Write the Java code for the algorithm?

Note: see example at the bottom of this page:

http://www-groups.mcs.st-and.ac.uk/~vince/teaching/summer04/flow.pdf

5. Read the following links on Genomics. Nothing to report except it would be on next quiz.
sSlides Attached
https://ghr.nlm.nih.gov/primer/basics/gene
https://www2.le.ac.uk/projects/vgec/highereducation/topics/dna-genes-chromosomes
https://ghr.nlm.nih.gov/primer/mutationsanddisorders/possiblemutations