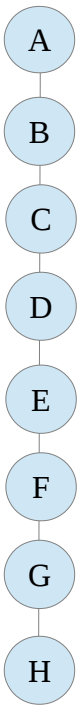
Joshua Rinaldi
101902285
10/8/14
HW4

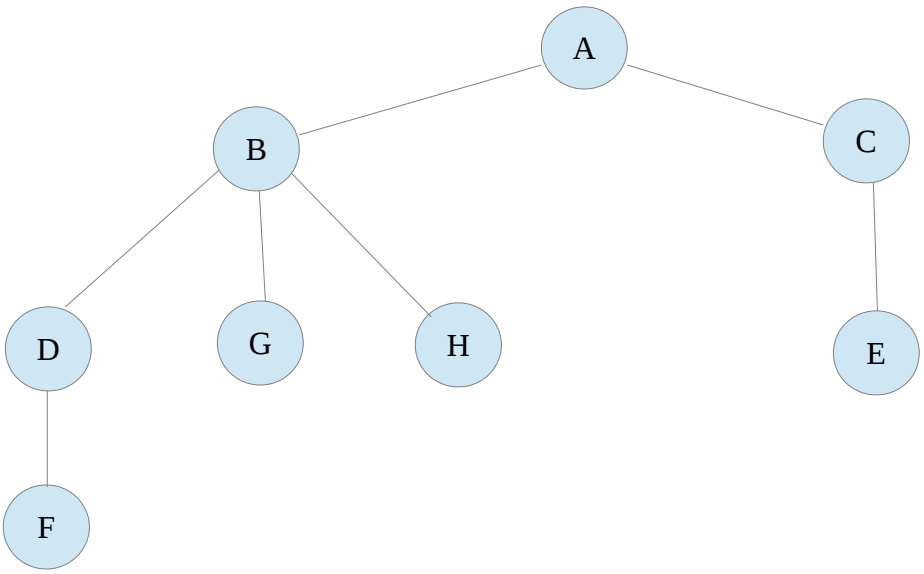
Honor Code Pledge:_____

Question 1

(a)

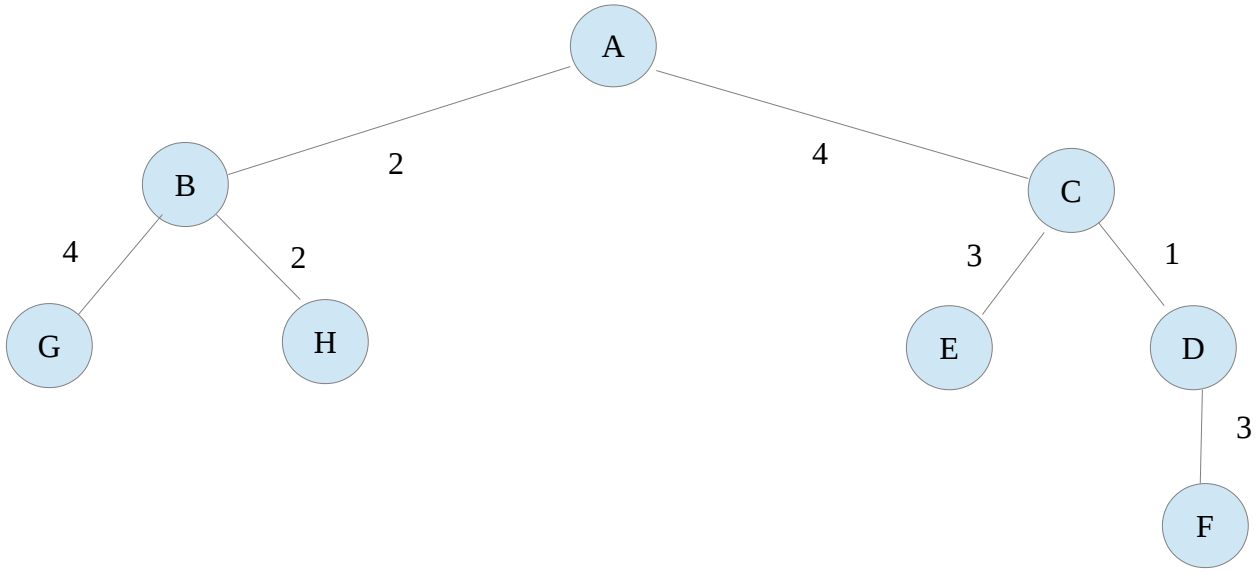


(b)



(c)

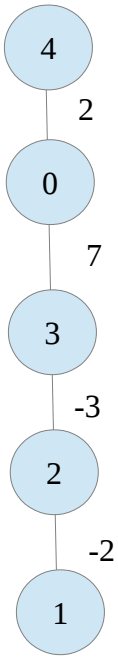
A	B	C	D	E	F	G	H
0	∞	∞	∞	∞	∞	∞	∞
0	2	4	∞	∞	∞	∞	∞
0	2	4	11	∞	∞	6	4
0	2	4	5	7	∞	6	4
0	2	4	5	7	∞	6	4
0	2	4	5	7	8	6	4
0	2	4	5	7	8	6	4
0	2	4	5	7	8	6	4
0	2	4	5	7	8	6	4



(d)

4	3	2	1	0
Iteration 1				
0	∞	∞	∞	∞
0	∞	7	∞	2
0	9	7	8	2
0	9	7	5	2
0	9	7	5	2
0	9	6	5	2
Iteration 2				

0	9	6	5	2
0	9	6	5	2
0	9	6	5	2
0	9	6	5	2
0	9	6	4	2
0	9	6	4	2
Iteration 3				
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
Iteration 4				
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2
0	9	6	4	2



Question 2

Because the graph is strongly connected and it has only positive edge lengths, the most efficient way to find the shortest path between any two nodes is to use Dijkstra's algorithm.

In the situation where the same graph is used with multiple different V_0, the most efficient way to find a path between A and B is to make an array that will store the path between every pair of nodes in the graph. You then search through the graph to find the shortest path between A and V_0 and the shortest path between V_0 and B. You then combine these two paths to come up with the path between A and B that will go through V_0. This would take $O(V^2*((|(V)|+|(E)|)*\log|(V)|))$ time to run.

Another, more efficient option would be to create two arrays. One of the arrays would store the path from every node A to V_0 and the other array would store the path from V_0 to every node B. You would then search through the arrays to find the corresponding paths for A and B, and then combine these paths. This would take $O(V*((|(V)|+|(E)|)*\log|(V)|))$ time to run.

The choice between algorithms comes down to whether or not there will be multiple V_0 for one graph. If this is the case, the second algorithm becomes considerably less efficient, as you need to keep creating arrays, whereas you only need to create one array in the first algorithm. However, because the run time to create the array(s) in algorithm one is $O(V^2)$ and in algorithm 2 is $O(V)$, if you are only using one V_0, it is more efficient to use the second algorithm.