CSCI 6470
Homework 5
Jiakuan Li

1.

(1) The MST is a subgraph of the full connected graph . The full connected graph connect all the vertexes and the edge obey the triangle inequality law. $w(u, v) \leq w(u, i) + w(i, v)$. So, if $w(u, v)$ is the shortest edges, it will definitely in the spanning graph. Since there is no better choice from u to v via another vertex, this will be the selected path while generating the MST.

(2) Yes, if the weight of all the edges are unique, then edge(x, y) will be in the MST. While constructing the MST, we order the edges from smallest to largest, if the edge does not make the cycle, we will included into the MST. Since all the edges are unique weighted, (u, v) will be the first edge included, (x, y) will be the second. And two edges could not form a cycle. Therefore, (x, y) will be in the MST if it is the second smallest edge and the weight of all edges are unique.

2.

If the edge $(u, v) \in T$, it is the smallest edge among the edges connected to vertex v. Any cut will separate the graph into two part, and the light edge is the edge with smallest weight that across the cut. Sine (u, v) is the relatively smallest, therefore, if the cut cross any of the (u, v), it will be the light edge.

3.

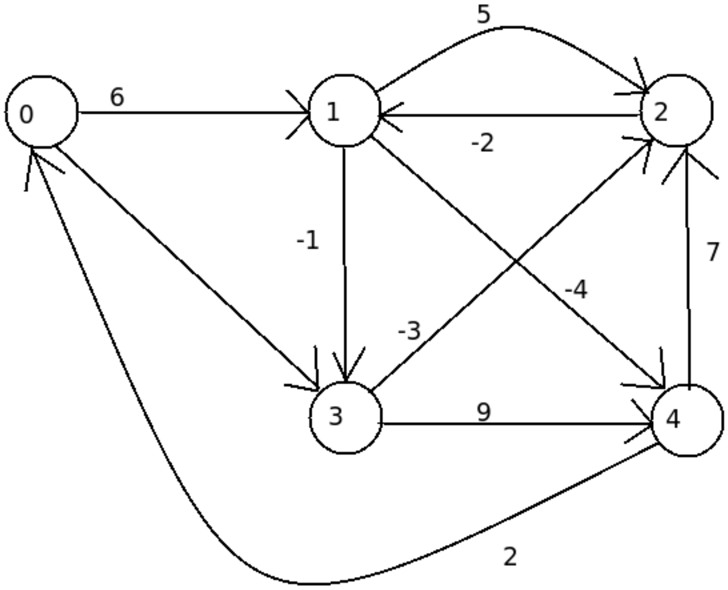
(1) Use the $FINDSET(U)$,$FINDSET(V)$ to identify if it is a edge in MST. If yes, the cut will cross this edge and the edge will be added to MST solution.

(2) When select the min edge, it will detect if it is connected to the old(visited) vertex. It store A using a linklist like data structure. For every vertex, it store the parent of the itself.

4.

(1)

generation	4	π	0	π	1	π	2	π	3	π
1^{st}	0	NULL	2	4	4	2	6	3	9	0
2^{nd}	0	NULL	2	4	4	2	6	3	9	0
3^{rd}	0	NULL	2	4	4	2	6	3	9	0
4^{th}	0	NULL	2	4	4	2	6	3	9	0



(2)

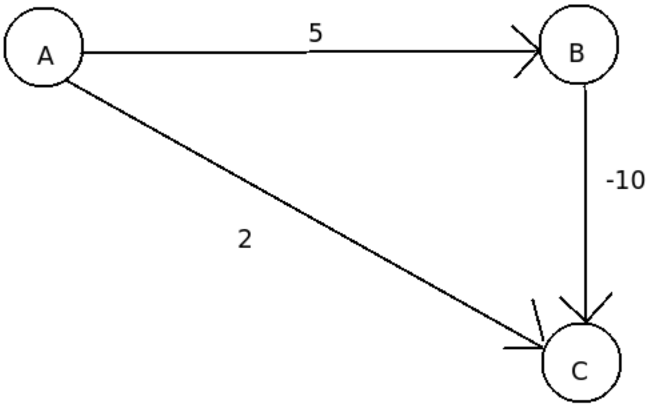
generation	4	π	0	π	1	π	2	π	3	π
1^{st}	0	NULL	2	4	2	2	4	3	7	3
2^{nd}	-2	1	2	4	-4	2	-2	3	1	1
3^{rd}	-8	1	0	4	-10	2	-8	3	-5	1
4^{th}	-14	1	-6	4	-16	2	-14	3	-11	1

After running the iteration, check weight of edges. Look at vertex 1, the $d(1) = -16$, $w(1,3) = -1$, and $d(3) = -11$. In BELLMAN-FORD algorithm, we know that if $d(v) > d(u) + w(u,v)$ that means a negative cycle. In this case, $d(v) = -11$, $d(u) = -16$, and $w(u,v) = -1$. $-11 > -16 + (-1)$, so there is a negative cycle in the graph.

5.

No. No matter how the input graph is, the outer for loop always run $|V| - 1$ times and the inner for loop runs for $|E|$ times. So, the time complexity for BELLMAN-FORD algorithm is $O(|V| - 1 \cdot |E|)$. But the DAG-SHORTESTPATH is $O(V + E)$.

6.



generation	A	π	B	π	C	π
1^{st}	0	NULL	5	A	2	A
2^{nd}	0	NULL	5	A	2	A
3^{rd}	0	NULL	5	A	-5	B

Dijkstra's algorithm works fine if the negative edge appear leaving the source vertex. Assume the vertex S has k out going edges towards k vertex, $(v_1, ..., v_k)$. Note that $v_i < v_j$, so there is no "better" path from S to v_i through v_j. So the order of investing does not change. Therefore, if the negative edges is leaving the source state, Dijkstra's algorithm works fine.

7.

To detect the negative cycle is easy. After running all iteration, check every vertex in G. If there exists a vertex that has a negative distance itself, that means there is a negative cycle in the graph G.