Suggested Solutions to HW #9

1. (7.6) Consider Algorithm *Single_Source_Shortest_Paths* (discussed in class). Prove that the subgraph consisting of all the edges that belong to shortest paths from v, found during the execution of the algorithm, is a tree rooted at v.

Solution.
We can conclude that the subgraph is a tree rooted at v from the following two facts.

(a) For every vertex $u \neq v$ in the subgraph, there is only one path from v to u in the subgraph.

(b) The subgraph is acyclic.

The fact 1(a) is trivial since a vertex u is added only if u was unmarked and (1) there is an edge (v, u) or (2) there is a shortest path in the subgraph from v to some vertex w and there is an edge (w, u). The fact 1(b) is ensured since a new unmarked vertex is added at each iteration of the algorithm. □

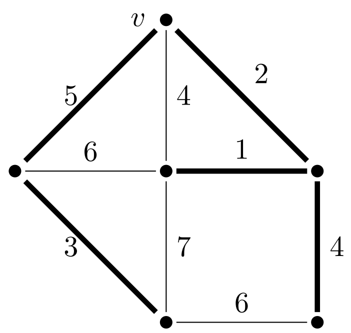
2. (7.12)

(a) Give an example of a weighted connected undirected graph $G = (V, E)$ and a vertex v, such that the minimum-cost spanning tree of G is the same as the shortest-path tree rooted at v.

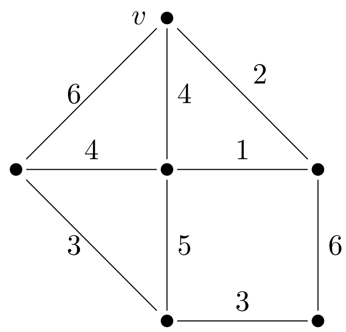
(b) Give an example of a weighted connected undirected graph $G = (V, E)$ and a vertex v, such that the minimum-cost spanning tree of G is very different from the shortest path tree rooted at v. Can the two trees be completely disjoint?

Solution.

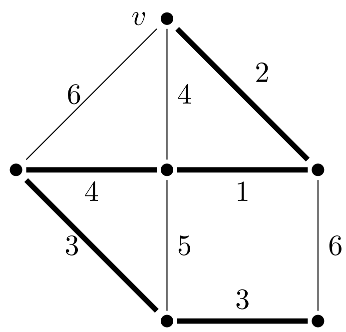
(a) The graph G is shown below. The minimum-cost spanning tree of G and the shortest path tree rooted at v are represented by thick edges.



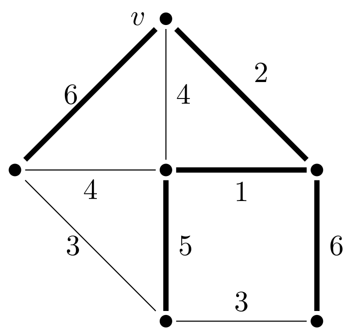
(b) The graph G is shown below.



The minimum-cost spanning tree of G is represented by the thick edges in the following graph.



The shortest path tree rooted at v is represented by the thick edges in the following graph.



It is impossible that the two trees are completely disjoint. Let T_m denote the minimum cost spanning tree of G and T_s be the shortest path tree rooted at v. Consider the following two cases: (1) v has only one edge; (2) v has more than one edge. In case (1), the only edge must be both in T_m and in T_s. Otherwise v will be disconnected from T_m and from T_s. In case (2), let (v, u) be the minimum weighted edge among all edges of v. Then (v, u) must belong to both T_m and T_s. If (v, u) is not in T_m, then T_m must include some other edge (v, w). By replacing (v, w) with (v, u) in T_m, we get a new tree with lower weight, which contradicts that T_m is the minimum-cost spanning tree. If (v, u) is not in T_s, then the shortest path π from v to u must include some other edge (v, w). Since (v, u) is lighter than (v, w), the edge (v, u) is shorter than π, which contradicts that π is the shortest path from v to u.

□

3. (7.16 modified)

(a) Run the strongly connected components algorithm on the directed graph shown in Figure 1. When traversing the graph, the algorithm should follow the given DFS numbers. Show the *High* values as computed by the algorithm in each step.

(b) Add the edge $(4, 2)$ to the graph and discuss the changes this makes to the execution of the algorithm.

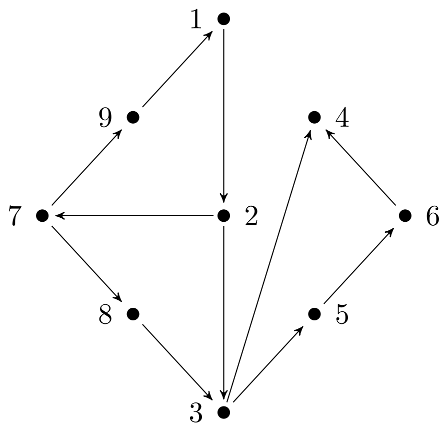


Figure 1: A directed graph with DFS numbers

Solution.

We use the DFS number as the vertex ID.

(a) The *High* values computed are shown below.

Vertex	1	2	3	4	5	6	7	8	9
DFS_N	9	8	7	6	5	4	3	2	1
1	9	-	-	-	-	-	-	-	-
2	9	8	-	-	-	-	-	-	-
3	9	8	7	-	-	-	-	-	-
4	9	8	7	6	-	-	-	-	-
3	9	8	7	6	-	-	-	-	-
5	9	8	7	6	5	-	-	-	-
6	9	8	7	6	5	4	-	-	-
5	9	8	7	6	5	4	-	-	-
3	9	8	7	6	5	4	-	-	-
2	9	8	7	6	5	4	-	-	-
7	9	8	7	6	5	4	3	-	-
8	9	8	7	6	5	4	3	2	-
7	9	8	7	6	5	4	3	2	-
9	9	8	7	6	5	4	3	2	9
7	9	8	7	6	5	4	9	2	9
2	9	9	7	6	5	4	9	2	9
1	9	9	7	6	5	4	9	2	9

(b) Adding the edge $(4, 2)$ will make all vertices belong to the same strongly connected component. The new *High* values computed are shown below.

Vertex	1	2	3	4	5	6	7	8	9
DFS_N	9	8	7	6	5	4	3	2	1
1	9	-	-	-	-	-	-	-	-
2	9	8	-	-	-	-	-	-	-
3	9	8	7	-	-	-	-	-	-
4	9	8	7	8	-	-	-	-	-
3	9	8	8	8	-	-	-	-	-
5	9	8	8	8	5	-	-	-	-
6	9	8	8	8	5	6	-	-	-
5	9	8	8	8	6	6	-	-	-
3	9	8	8	8	6	6	-	-	-
2	9	8	8	8	6	6	-	-	-
7	9	8	8	8	6	6	3	-	-
8	9	8	8	8	6	6	3	7	-
7	9	8	8	8	6	6	7	7	-
9	9	8	8	8	6	6	3	7	9
7	9	8	8	8	6	6	9	7	9
2	9	9	8	8	6	6	9	7	9
1	9	9	8	8	6	6	9	7	9

□