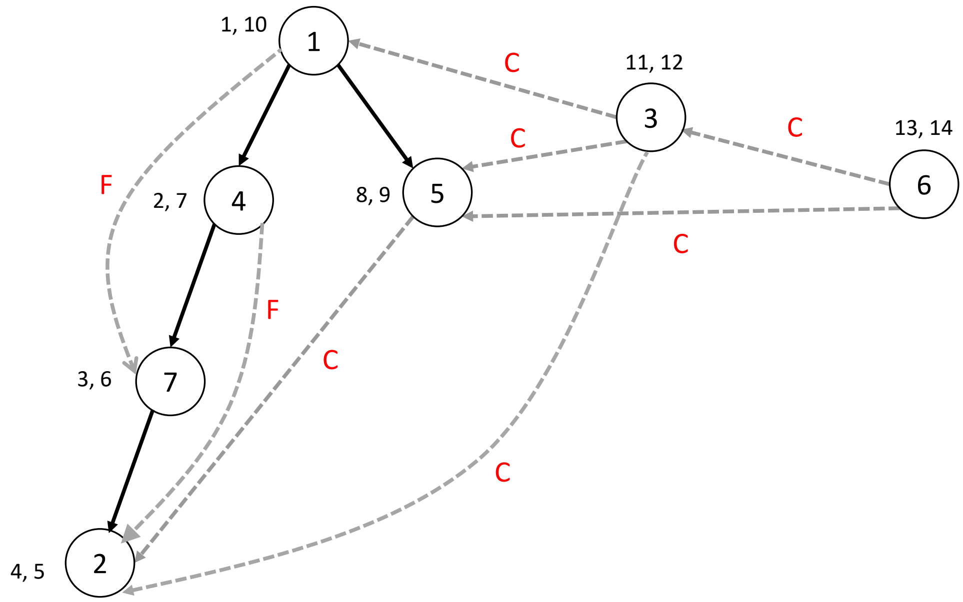
Solutions for Homework Assignment #6

Answer to Question 1.

a.



b. The above DFS has 0 back edges, 2 forward edges, and 6 cross edges.

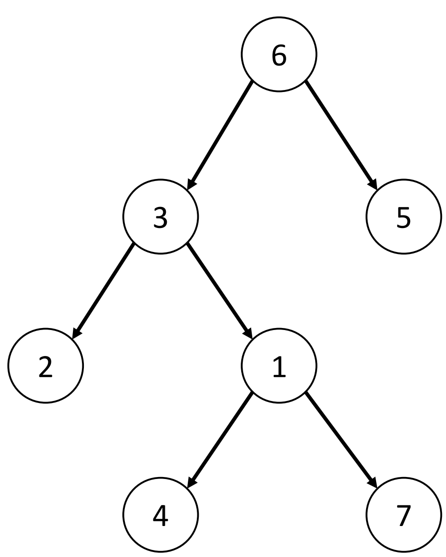
c. By part (b), some DFS of G has no back-edges. In class we proved that:

Theorem: For every directed graph G and every DFS of G, G has a cycle iff the DFS of G has a back edge.

Thus, G has no cycles. Therefore there is a topological sort of G, i.e., the courses can be taken in an order that satisfies all the prerequisites.

d. The topological sort algorithm outputs all the nodes of G in order of decreasing $f[]$ "finish" times. This gives the following list: 6, 3, 1, 5, 4, 7, 2.

e. Draw a Breadth-First Search tree of G that ***starts at node* 6** and explores the edges in the order of appearance in the above adjacency lists.



Answer to Question 2.

1. This claim is false. Here's a counter-example.

   Let $G = (V = \{1,2,3,4\}, E = \{(1,2),(1,3),(1,4),(2,3),(2,4)\})$, a connected, undirected graph with distinct edge weights $\{w((1,2)) = 1, w((2,3)) = 2, w((1,3)) = 3, w((2,4)) = 4, w((1,4)) = 5\}$. Then $T_1 = (V, E_1 = \{(1,2),(2,3),(1,4)\})$ is an MST with weight $w(T_1) = 7$. But now $T_3 = (V, E_3 = \{(1,2),(1,3),(2,4)\})$ and $T_4 = (V, E_4 = \{(1,2),(2,3),(1,4)\})$ have $w(T_3) = 8 = w(T_1) + 1 = w(T_4)$, but $T_3 \neq T_4$.

2. This claim is true. Here's a proof.

   Let $G = (V, E)$ be a connected, undirected graph, with positive edge weights. Let $E' \subseteq E$ and $G' = (V, E')$ be a spanning subgraph of G of minimum weight.

   Assume, for the sake of contradiction, that G' is not an MST. Since G' has minimum weight among spanning subgraphs of G, including subgraphs whose edge sets are MSTs, it must fail to be an MST for the only other possible reason: by not being a tree. So G' contains a cycle C. Let e be an edge in C with weight $w(e) > 0$. Then $G'' = (V, E' - \{e\})$ is also a spanning subgraph of G but has weight $w(G'') = w(G') - w(e) < w(G')$ $\rightarrow\leftarrow$ Contradiction: G' has minimum weight among spanning subgraphs of G. ■

3. This claim is false. Here's a counter-example.

   Let $G = (V, E)$ be the complete graph on $|V| = n \geq 3$ vertices (hence $|E| = n(n-1)/2$), with all edges having weight 0. Then G and all connected subgraphs of G have minimal weight: 0, even though E itself is not an MST.

Answer to Question 3.

a. Let $G = (V, E)$ be a connected, undirected graph representing the lakes in L, and weighted edges representing portages in P, with the weights being the length of each portage. By assumption, all the weights of edges in E are distinct, and graph $G = (V, E)$ is stored as adjacency list P.

b. Let $G = (V, E)$ be a finite, connected, undirected graph with each edge in E having a distinct weight. Assume G contains cycle $C = u = v_0, v_1, \ldots, v_k = v, u$, where $k > 1$ and edge (u, v) is heavier than any other edge in C. Let $P = t_0, \ldots, t_j$ be a minimum toughness path in G there are two cases to consider.

case (u, v) is not an edge in P: Then removing edge (u, v) does not change the fact that P is a minimum toughness path.

case (u, v) is an edge in P: Then $u = t_i$ and $v = t_{i+1}$ for some adjacent vertices t_i, t_{i+1} in P. But then the walk[1] W' that replaces edge (u, v) in P by $u = v_0, v_1, \ldots, v_k = v$ contains path P' from t_0 to t_j and has toughness no greater than P, and hence must be a minimum toughness path also.

In both possible cases there remains a minimum toughness path between an arbitrary pair of vertices.

c. Let $G = (V, E)$ be a finite, connected, undirected graph with each edge in E having a distinct weight. Let $E' \subseteq E$, let $T = (V, E')$ be an MST for G, let i, j be a pair of distinct vertices in V, and let P be the path in T from i to j. Assume, for the sake of contradiction, that P is not a minimum toughness path in G, so the rating of (i, j) is higher in T than in G. Thus there must be a minimum toughness path $P' \neq P$ from i to j in G. Let (x, y) be the maximum weight edge in P, so all edges in P' must be lighter than (x, y).

Removing (x, y) partitions T into two connected components, one containing i and the other j. Minimum toughness path P' must have an edge (x', y') across this cut, and the weight of (x', y') must be less than the weight of (x, y). Exchanging (x, y) for (x', y') creates a spanning tree $T' = (V, (E' - \{(x, y)\}) \cup \{(x', y')\})$, which has lower total weight than T. This contradicts the choice of T, so the assuumption that P is not a minimum toughness path is false. So T preserves the rating between every pair of vertices in V.

[1] In graph theory a walk W is a sequence of vertices $w_0, \ldots, w_k$ where adjacent pairs of vertices have an edge between them. Much like a path, but repeated vertices are allowed.