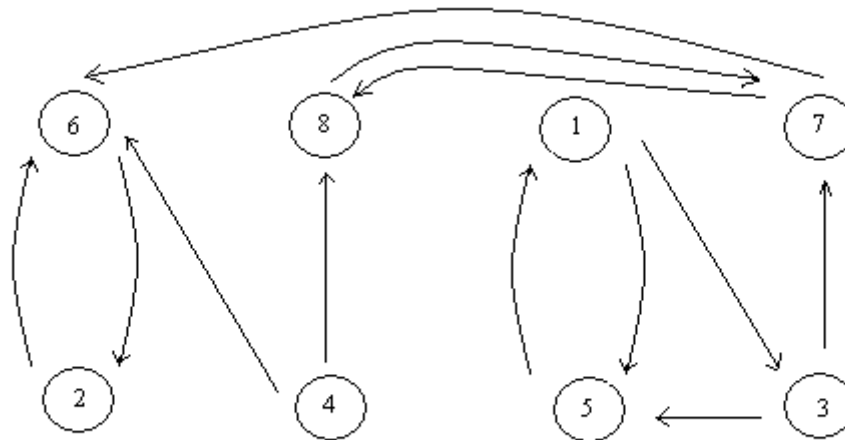


Problem 1 (20 Points)

a)

Run the Strongly Connected Components (SCC) algorithm on the following graph. Show the pre and post numbers for both passes of DFS. Where there are several choices of vertex to start at or edges to follow, go to the vertex that occurs earliest numerically.



b)

How can we topologically sort the SCCs without running another pass of DFS? Assume you still have all the information generated by the SCC algorithm available, including pre and post numbers generated during each pass of DFS, and the SCC number assigned to each vertex.

Problem 2 (24 Points)

True or False? No explanation required except for partial credit. Each correct answer is worth 2 points, but 2 points will be subtracted for each wrong answer, so answer only if you are reasonable certain.

1. Let G be a directed graph. The Bellman-Ford algorithm will find the longest path (if it is finite, or report if it is not) from the start vertex s to any other vertex v in graph if we negate all the edges weights $W(e)$ before running the algorithm.
2. Let G be a directed graph with positive edge weights $W(e)$. Let $W_{\max} = \max_e W(e)$ be the maximum edge weight. Dijkstra's algorithm will find the longest path from the start vertex s to any other vertex v in a graph if we replace all the edges weights $W(e)$ by the new positive weights $1 + W_{\max} - W(e)$ before running the algorithm.
3. $e^{c\sqrt{v}}$ is $O(e^{\sqrt{v}})$ for all $c > 0$.
4. $n^{2 \sin(n)/\log(n)}$ is $O(n^2)$.
5. After we run DFS on a directed graph G , it is possible for a single vertex $v \in G$ to be simultaneously incident on at least one back edge, forward edge, cross edge and tree edge.
6. It is always cheaper to use a binary heap instead of a linked list to implement Dijkstra's algorithm.
7. Let X be a minimum spanning tree of the weighted undirected graph $G(V, E)$. Let $G'(V', E')$ be another graph defined by augmenting G with a new vertex u and some weighted edges incident on u . In other words, $V' = V \cup \{u\}$ and $E' = E \cup F$ where F is a set of edges touching u . Then there is always a minimum spanning tree X' of G' such that $X \subset X'$.
8. Let $G(V, E)$ be an undirected graph. Then G has a cycle if and only if it is possible to put directions on all the edges in E so that every vertex in V has an edge pointing to it.
9. The paths computed by Bellman-Ford after the 2nd iteration might be the shortest.
10. Let $T(0) = 1$ and $T(n) = 2^{T(n-1)}$ for $n \geq 1$. Then the cost of computing $T(n)$ and printing it out as a binary integer is $\theta(T(n-1))$.
11. If $T(n) = 49 T(n/7) + n^2 \cos\sqrt{n}$, then $T(n) = O(n^2 \log n)$.
12. Let G be an undirected graph, and let G' be another undirected graph. G' has one vertex for each biconnected component of G (and no other vertices). Two vertices in G' are connected by an edge if and only if the two corresponding biconnected components in G share a vertex. Then G' can have a cycle.

Problem 3 (18 Points)

Let $G(V, E)$ be a directed graph with a weight $W(v)$ associated with each vertex $v \in V$. Let $p = \langle v_0, v_1, v_2, \dots, v_n \rangle$ be a path in G , with no repeated vertices. Then the weight of the path p is defined as the sum of the vertex weights on the path: $W(p) = \sum_{i=0}^n W(v_i)$.

a) Suppose the vertex weights are positive. Let $s \in V$ be a vertex. Give an efficient algorithm for finding the paths of minimum weight from s to all other vertices in G . You should give high-level pseudo code for the algorithm, justify its correctness, and analyze its complexity in a $O()$ sense. You may use any algorithm presented and analyzed in class so far without repeating its correctness proof or complexity analysis.

b) Now suppose the vertex weights may take any values, not just positive ones, and answer the same question as in part a.

Problem 4 (18 Points)

Let T be a MST in the weighted graph $G(V, E)$. Let $G'(V', E')$ be a graph obtained by removing one vertex u from G (so that $V' = V - \{u\}$), and by removing all the edges incident on u .

Let $e \in E$ any edge of T not incident on u . Prove that there is an MST of G' that contains e . Hint: the cut property says:

Let $X \subseteq T$ where T is a Minimum Spanning Tree (MST) in the weighted graph $G(V, E)$. Let $S \subset V$ such that no edge in X crosses between S and $V - S$; i.e. no edge in X has one endpoint in S and one endpoint in $V - S$. Among the edges crossing between S and $V - S$, let e be an edge of minimum weight. Then $X \cup \{e\} \subseteq T'$ where T' is a MST in $G(V, E)$.

Partial Answers to Problem 2

1. True
2. False
3. False
4. True
5. True
6. False
7. ?
8. False
9. True
10. True
11. True
12. True