50.004 Introduction to Algorithms, Fall 2015
Singapore University of Technology and Design — 21 Oct., 2015
Simon Lui — Home Exercises - Week 6

Student Information

Name: Zhang Zhexian Student ID: 1001214
Due Date: 2-Nov. 4pm.
Submit written answer on paper in class or submit electronic version online through Dropbox. Submission without student information will **NOT** be marked!

Note: Exercise 1 only has 1 answer. The other questions CAN have multiple answers.

Exercise 1

The adjacency list representation is better in terms of space then the matrix representation for graphs that are not dense (i.e., the number of edges $m = O(n)$, where n is the number of nodes) because C

A. It is simpler to represent the edges from each node as a linked list.

B. An adjacency matrix representation does not explicitly name the set of edges.

C. When the graph is not dense, the number of edges is $m = O(n)$ and the space complexity in bits when using the adjacency list representation is $O(n \log n)$ instead of $O(n^2)$.

D. For non-dense graphs both representations are equivalent since the complexity is proportional to the number of nodes.

Exercise 2

In the fist BFS algorithm, a frontier set at some stage of the computation stores B, E

A. All the nodes visited so far.

B. A set of nodes from which we like to determine the reachable nodes in one step that are new. ✓

C. All the nodes that can be reached in k steps from the source node, for some appropriate k.

D. All the nodes that can be reached at least in k steps from the source node, for some appropriate k .

E. All the nodes for which the minimum number of steps to be reached from the source node is exactly k, for some appropriate k . ✓

Exercise 3

In the BFS implementation, the frontier (or array) + Adj list implementation and the Queue + Adj list implementation, assuming that adjacency information $\mathrm{Adj}[u]$ is implemented in the same way in both algorithms (same sequence of successor nodes for each node), then A , E

A. The two algorithms generate potentially different BFS trees.

B. The two algorithms generate same BFS trees.

C. Each of the two algorithms generates always a different tree but the level sets of nodes are the same.

D. Each algorithm eventually visits all the nodes in V.

E. Each algorithm visits only the nodes reachable from the source node. ✓

Exercise 4

In the DFS which of the following properties are true A C D F G

A. Grey nodes form a stack where new nodes are added (pushed) at the top and are also reduced (popped) from the top. ✓

B. Tree edges are edges from black or grey nodes to white nodes.

C. A black node can have unexplored edges only to grey nodes. ✓

D. Two nodes u, v for which $u.d, u.f < v.d, v.f$ are in different subtrees of the DFS tree. ✓

E. Two nodes u, v for which $u.d < v.d < v.f < u.f$ are in the same subtree of the DFS tree and v is an ancestor of u. ×

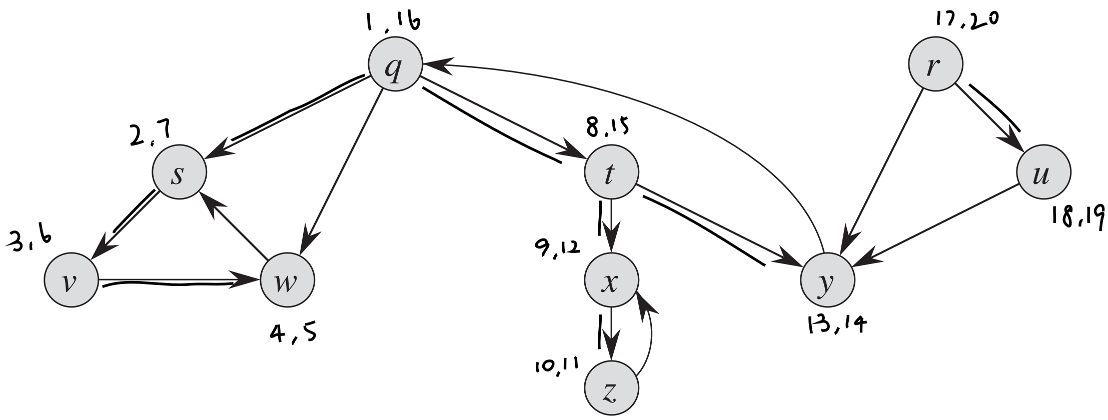
F. The nodes of the DFS forest may correspond to a strict subset of the nodes of the graph. ✓

G. If a directed graph contains a path from u to v, then any DFS will result in $v.d \leq u.f$ ✓

H. If we use our algorithm of topological sort on a general directed graph, it will always terminate but the answer will be wrong if there are cycles.

Exercise 5

Consider the graph below, assume that in the DFS algorithms in $\mathrm{DFS(G)}$ the nodes are visited in alphabetical order, and in $\mathrm{DFS-Visit(G,u)}$, the $Adj[u]$ list is also ordered alphabetically (we visit the neighbors of u in alphabetical order). Which of the following are true A , D , F



A. The discovery/finishing times for some of the nodes are $q = (1, 16), r = (17, 20), y = (13, 14)$. ✓

B. The discovery/finishing times for some of the nodes are $q = (1, 18), r = (19, 20), z = (21, 22)$. ✗

C. The edges $(v, w), (z, x), (q, w), (u, y)$ are respectively forward, backward, forward, cross edges. ✗

D. The edges $(v, w), (z, x), (q, w), (u, y)$ are respectively tree, backward, forward, cross edges. ✓

E. If there was an edge (y, w) , it would be a backward edge. ✗

F. If there was an edge (y, w) , it would be a cross edge. ✓