

## CSE 830: Homework #4

Due: Thursday, November 3<sup>rd</sup> 2011, 10:20am

1. In breadth-first and depth-first search, an *undiscovered* node is marked *discovered* when it is first encountered, and marked *completely-explored* when it has been searched. At any given moment, various nodes can be in any of these states. In all cases below, describe a graph on  $n$  vertices with a particular starting vertex  $v$  that has the properties described, and draw an example of it.
  - a. At some point during a breadth-first search,  $\Theta(n)$  are simultaneously in the discovered state.
  - b. At some point during a depth-first-search  $\Theta(n)$  nodes are simultaneously in the discovered state.
  - c. At some point during a breadth-first-search  $\Theta(n)$  nodes are in the undiscovered state,  $\Theta(n)$  nodes are in the completely explored state, and only  $\Theta(1)$  nodes are discovered.
  - d. At some point during a depth-first-search  $\Theta(n)$  nodes are in the undiscovered state,  $\Theta(n)$  nodes are in the completely explored state, and only  $\Theta(1)$  nodes are discovered.
2. An *articulation vertex* of a graph  $G$  is a vertex whose deletion disconnects  $G$ . Let  $G$  be a graph with  $n$  vertices and  $m$  edges.
  - a. Give a simple  $O(n+m)$  algorithm for finding a vertex of  $G$  that is *not* an articulation vertex, i.e. whose deletion does not disconnect  $G$ .
  - b. Now expand this algorithm to one that finds a deletion order for the  $n$  vertices, also in  $O(n+m)$  time, such that no deletion disconnects the graph. (Hint: Think BFS or DFS).
3. The *Maximum Clique* of a graph  $G$  is the largest sub-graph where all pairs of vertices in that sub-graph have an edge connecting them; it is known to be a hard problem. A brute force approach might test all possible sub-graphs. Describe optimizations to solve maximum clique as fast as possible. Specifically, consider backtracking, bounding, the order in which the vertices are tested, and polynomial-time simplifications. You do not have to write out the full algorithm, just describe the optimizations. Your optimized algorithm MUST always return the correct (maximum) answer.
4. An *Euler tour* of a strongly connected, directed graph  $G = (V; E)$  is a cycle that traverses each edge of  $G$  exactly once, although it may visit a vertex more than once. Describe an  $O(E)$ -time algorithm to find an Euler tour of  $G$  if one exists.
5. A *tournament* is a directed graph formed by taking the complete undirected graph and assigning arbitrary directions on the edges, i.e. a graph  $G = (V; E)$  such that for all  $u$  and  $v$  in  $V$ , exactly one of  $(u; v)$  or  $(v; u)$  is in  $E$ . Show that every tournament has a Hamiltonian path, that is, a path that visits every vertex exactly once. Give an algorithm to find this path
6. (*Extra Credit*) Two members of the Lansing bomb squad, E. Fudd and Y. Sam, are called to the scene of a bomb scare. They study the bomb and figure out an order to cut the wires, but want to move as little as possible to minimize their chance of accidentally setting it off. Given that there is a distance  $d_{i,j}$  between any pair of wires  $i$  and  $j$ , and that Mr. Fudd cuts the first wire and Mr. Sam cuts the second, give a polynomial-time algorithm to determine who should cut each of the remaining  $n-2$  wires to minimize the total distance moved. (*Hint*: Think dynamic programming!) Analyze the time complexity for this algorithm and draw a diagram of any tables used.