CSCI-GA.1170-001/002 Fundamental Algorithms December 3, 2015

Solutions to Problem 4 of Homework 10 (6 points)

Name: GOWTHAM GOLI (N17656180) *Due: Tuesday, December 1*

Give an algorithm that determines whether or not given a undirected graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(V)$ time, independent of $|E|$. Make sure you prove the correctness of your algorithm.

Solution:

Algorithm

There will be a cycle in G if in the DFS of G, any unexplored edge visits a node that has already been visited earlier i.e this unexplored edge is a backward edge in the DFS forest of G. Therefore if the DFS yields no back edges i.e it contains only tree edges then G is acyclic or else G contains a cycle

Correctness and Running Time

If the given graph G is acyclic then the DFS forest will not contain any egde (u, v) such that v is an ancestor of u. Suppose there exist such an edge (u, v) then $DFS(v)$ will visit u as v is the ancestor of u and then $DFS(u)$ visits v which is already visited thus resulting in a cycle. Therefore it contradicts our assumption and hence the edge (u, v) can't exist i.e DFS yields no back edges and contains only tree edges if G is acyclic

Therefore if G is acyclic, then maximum number of tree edges a graph can have is at most $|V| - 1$. Therefore a single run of DFS is sufficient to check for back edges in this case. Therefore the running time will be $O(m + n) = O(2n) = O(n)$

Suppose that G contains a cycle, then DFS of G will contain at least one back edge. Note that while running DFS this back edge can be found before seeing $|V|$ edges. Therefore there are only $O(|V|)$ number of operations and the back edge will be found before that. Hence the running time will be $O(n)$. Therefore the algorithm will run in $O(|V|)$, independent of $|E|$

□