Design and Analysis of Algorithms
Homework 6

Clearly number your solution to each problem. Staple your solutions and bring them to class on the due date. Express your algorithms in pseudo-code when directed. Always provide justification for your answer when asked to give the running time of an algorithm. Be brief and concise, and draw pictures where appropriate.

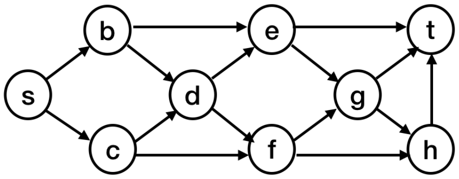
1. (5pts) Would you prefer DFS or BFS (or both equally) for the following tasks? Justify your answer. Assume the graph is undirected and connected.
   (a) Determine if the graph is acyclic.
   (b) Find a path to a vertex known to be near the starting vertex.
   (c) Find the connected components of the graph.

2. (5pts) A graph is **triconnected** if one has to remove at least 3 vertices from the graph to disconnect it. Construct examples of the following graphs or explain why it cannot be done. Assume the graph is undirected.
   (a) A triconnected graph with exactly 5 vertices and 8 edges.
   (b) A triconnected graph with exactly 5 vertices and 6 edges.
   (c) A triconnected graph with exactly 8 vertices and 14 edges.

3. (10pts) A company named RT&T has a network of n switching stations connected by m high-speed communication links. Each customer's phone is directly connected to one station in his or her area. The engineers of RT&T have developed a prototype video-phone system that allows two customers to see eachother during a phone call. In order to have acceptable image quality, however, the number of links used to transmit video signals between the two parties cannot exceed 4. Suppose that RT&T network is represented by a graph. Design and give the pseudo-code for an efficient algorithm that computes, for each station, the set of stations it can reach using no more than 4 links. Analyze its running time.

4. (10pts) An **Eulerian cycle** of a directed graph G with n vertices and m edges is a cycle that traverses each **edge** of G exactly once according to its direction. Such a cycle always exists if the in-degree is equal to the out-degree for each vertex in G. Describe in pseudo-code a $O(n + m)$ time algorithm for finding an Euler tour of such a graph G. Analyze its running time.

5. (10pts) Trace the execution of **TopologicalSort** algorithm (as given on page 326) on the following graph.



To review, here is the pseudo-code for the algorithm:

```
Input: A digraph $G$ with $n$ vertices
Output: A topological ordering $v_1, v_2, ...v_n$ of $G$
function TopologicalSort($G$)
    Let $S$ be an initially empty stack
    for each vertex $u$ of $G$ do
        Let incounter($u$) be the in-degree of $u$
        if incounter($u$) $= 0$ then
            $S$.push($u$)
    $i \leftarrow 1$
    while $S$ is not empty do
        $u \leftarrow S$.pop()
        Let $u$ be vertex number $i$ in the topological ordering
        $i \leftarrow i + 1$
        for each outgoing edge $e = (u, w)$ of $u$ do
            incounter($w$) $\leftarrow$ incounter($w$) $- 1$
            if incounter($w$) $= 0$ then
                $S$.push($w$)
    if $i > n$ then
        return $v_1, v_2, ...v_n$
    else
        return "digraph $G$ has a directed cycle"
```

Show the graph after each iteration of the while loop, and display the incounter and the currently assigned topological sorting labels at each one of these iterations.

6. (10pts) Bob loves foreign languages and wants to plan his course schedule to take the following nine language courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169. The course prerequisites are:

- LA15: (none)
- LA16: LA15
- LA22: (none)
- LA31: LA15
- LA32: LA16, LA31
- LA126: LA22, LA32
- LA127: LA16
- LA141: LA22, LA16
- LA169: LA32

Find a sequence of courses that allows Bob to satisfy all the prerequesites.