

CSE 326
Autumn 2005
Assignment 6
Due 11/16/05

For all algorithm and data structure design problems please provide elegant pseudocode and an adequate explanation of your methods. It is often helpful to include small examples demonstrating the method. Put your name at the top of each sheet of paper that you turn in.

1. A bipartite graph is one where there is a binary labeling of the vertices with the property that there is no edge between vertices with the same label. There is a famous theorem that states that a graph is bipartite if and only if it has no cycle of odd length. Design an algorithm based on depth-first search to determine if a graph is bipartite and if it is not return an odd length cycle in the graph. Your algorithm should use the adjacency list representation of a graph. Your algorithm should run in linear time. Hint: there are two labels for marking 1 and 2. A new vertex visited from a vertex marked 1 is marked 2 and a new vertex visited from a vertex marked 2 is marked 1.
2. Some project planning applications use a labeled acyclic directed graphs to represent the jobs and job times on a project. A vertex in the graph represents a job and its label represents the time the job will take. A directed edge from one vertex to another represents the fact the job represented by the first vertex must be completed before the job represented by the second vertex. Assume we have a directed acyclic graph $G = (1, 2, \dots, n, E)$ with vertices labeled by non-negative integers c_1, c_2, \dots, c_n . The label c_i represents the time job i will take. Assume further that every vertex is reachable by some path from vertex 1, vertex 1 has in-degree 0, vertex n is reachable by some path from every vertex, and n has out degree 0. Vertex 1 represent the beginning of the project and vertex n represent the end of the project. The length of a path from 1 to n is the sum of the labels on the vertices along the path. Design an algorithm based on the topological sort algorithm to find the length of a longest path from 1 to n in the graph. The length of the longest path represents how long the entire project will take. Sometimes a longest path is called a critical path. Your algorithm should use the adjacency list representation of a graph. The labels can be stored in an additional array. Your algorithm should run in linear time. Hint: ultimately you will need to compute the length of the longest path from 1 to every other vertex. In the topological sort, when a vertex achieves in-degree 0, the length of the longest path from 1 to it should be known.