CSE 413, Analysis of Algorithms

Fall Semester, 2004

Assignment 1: The Hamiltonian Cycle Problems

Due Date: Sept. 8, 2004 (at the beginning of CSE 413 class)

1. Exercise 1.9, page 6 of the textbook. (This is in fact an instance of the **Hamiltonian cycle problem**; also see pages 243-244 for the definition of this problem.) **(20 points)**

2. Design an algorithm to solve the Hamiltonian cycle problem for an arbitrary undirected graph (not necessarily dense), and analyze the time complexity of your algorithm. The input is an undirected graph represented by an adjacency matrix (see page 84 for the adjacency matrix representation of a graph). The output is either a sequence of vertices corresponding to a Hamiltonian cycle in the graph (if one exists), or a "No" if the input graph does not admit any Hamiltonian cycle. **(20 points)**

3. Write and test a computer program for **Problem 2** in either **C** or **C++** (**Java** would be fine too). For this exercise, submit your program source code and your output on some test examples. Also, run your program on graphs of 5, 6, 7, 8, 9, and 10 vertices. Record the running time of your program. Use Excel, or a similar tool, to plot the running time for each graph size. That is, you should plot a curve for the running time of your program on these graph sizes, and submit the curve. **(20 points)**

4. It is well known that the Hamiltonian cycle problem is NP-complete, which implies that it is very likely that efficient algorithms (i.e., those running in a polynomial time) for this problem do not exist. Thus, people tend to look for good **approximation algorithms** for solving the problem. In this exercise, we relax the constraint of the Hamiltonian cycle problem as follows: We still seek a cycle that contains every vertex of the input graph; however, such a cycle need not be *simple*, which means that some vertices can appear in the cycle multiple times. Furthermore, we restrict that the sought cycle must contain no more than $2n$ vertices for any input graph of n vertices. Such a cycle can be considered as an approximation solution for the Hamiltonian cycle problem with an approximation ratio of 2. Design a polynomial time algorithm for computing an **approximate** Hamiltonian cycle as defined above, and analyze the time complexity of your algorithm. **(20 points)**

   (**Hint**: Consider doing a depth-first search (pages 190-194) in the input graph and see what you can make out of it.)

Total Points: 80