Homework 8

Due Monday, August 10, at 9 AM (GMT-7)

CSc 345 – Summer 2014

Instructor: Qiyam Tung

Instructions

1. This is an individual assignment. You must do your own work.
2. If you are having difficulty and need to ask a question you can:
   (a) Ask questions in class.
   (b) Stop by my office hours (or make an appointment).
   (c) Post a question on Piazza.
   (d) Post a private question on Piazza if the question is too specific.
3. Show all work. Incomplete solutions will **not** receive full credit
4. You may write your solutions by hand, or you may type them using any appropriate program such as Microsoft Word, OpenOffice Writer, LaTeX, etc...
   However, the final copy should be in PDF form and formatted so that it is legible.
5. If the listed problem is only a number, refer to the online book for the description of the problem (starting at page 46).
6. The algorithm families are:
   - Randomized algorithms
   - Approximation Algorithms
   - Divide and Conquer
   - Backtracking
   - Greedy Algorithms
   - Dynamic Programming
   - Linear Programming
   - Genetic Algorithm

1 Problems (52 pts)

1. (4) Run topological sort on the graph in Figure 1

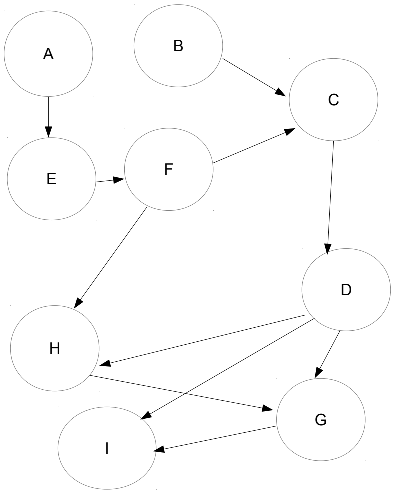

Figure 1: An unweighted directed graph.

2. (12) Exercise 16.3 on page 532 [Floyd's algorithm]. Show all the intermediate matrices computed as well as the final matrix.

3. (8) Write a dynamic programming solution for finding the Fibonacci number.

   (a) (2) Write down the recursive solution

   (b) (2) You will use an array to store your previously computed values. Describe what these values would be: for example, what does $F[3]$ represent?

   (c) (4) Write down your algorithm for finding the Fibonacci solution. Be sure to describe how you initialize your array and how you use the solutions to subproblems to find the solution to the whole problem. Note that you will receive no points if the solution is recursive.

4. (8) Write a dynamic programming solution for finding the $\binom{n}{r}$, given that that $\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}$.

   (a) (2) Write down the recursive solution

   (b) (6) Write down your algorithm for finding the solution. Be sure to describe how you initialize your table and how you use the solutions to subproblems to find the solution to the whole problem. Note that you will receive no points if the solution is recursive.

5. (8) Indicate which family of algorithms would be appropriate for solving the following problems and explain why it suits the problem. If it is an optimization problem, be specific about which algorithm family to use (i.e. genetic algorithm, linear programming, etc.).

   (a) (2) Making change with denominations 25, 16, 10, 5, and 1. Note that the problem can be stated as using as few coins as possible. In other words, Coins $= c_1 + c_2 + c_3 + c_4 + c_5$ (where c_1 is the number of quarters, c_2 is the number of 16 cent pieces, etc.). And, of course, the coins must add up to the change: $25c_1 + 16c_2 + 10c_3 + 5c_4 + c_5 =$ Change.

   (b) (2) Identifying the 3D structure of a plant in a 2D image. The solution space includes the position of the root (a point in a 100 × 100 × 100 box) the camera parameters (position, orientation), and the orientation of 10 branches and 20 leaves (orientation consists of integral numbers from 1 to 360).

   (c) (2) Finding the shortest route that visits every city once and returns to the original starting city? The application for solving such a problem appears in logistics, where solving it in a reasonable amount of time is important.

   (d) (2) Solving a maze. At a junction, you have at most 3 choices: left, forward, or right.

6. (4) Design a DFA that accepts the language

$$\{\text{w} \mid \text{w begins with “c” and ends with “ly”}\}$$
$$\Sigma = \{\text{c, l, y, a}\}$$

7. (2) Design a DFA that accepts the same language but has a different number of states.

8. (2) Design a regular expression that describes the language described by the DFA.

9. (4) Design a DFA that accepts binary strings such that the number of ones in the string are divisible by three For example, `111`, `011100`, and 00110010001101 are all strings that are accepted by the DFA. `1111`, `0001`, and `110` are strings that are not accepted by the DFA.