COMP 355: Advanced Algorithms

Practice Problems for Midterm 1

Midterm 1 will be on Wednesday, September 27th. The exam will be closed-book and closed-notes, but you will be allowed one cheat-sheet (front and back).

Disclaimer: These are practice problems. They do not necessarily reflect the actual length, difficulty, or coverage for the exam. You should also be sure to look over your homework problems as well as the in-class practice problems.

1. You should expect one problem in which you will be asked to work an example of one of the algorithms we have presented in class on a short example.

2. **Short answer questions.**
   a. Consider the code a = <0>, b = <01>, c = <11>, d = <101>. Is this a prefix code? Explain.

   b. As a function of n, give the (tight) asymptotic running time of the following three nested loops using Θ-notation. Briefly justify your answer.
   ```
   for (i = 1 to n)
      for (j = 1 to n - i)
         for (k = j to i + j)
            ...something taking constant time...
   ```

   c. Recall that in the interval scheduling problem, you are given n requests, each having a start time s_i and finish time f_i and the objective is to schedule the maximum number of non-conflicting tasks. Which of the following greedy strategies is optimal? (List all that apply. No explanation needed.)
      i. Earliest start time first (select in increasing order of s_i)
      ii. Earliest finish time first (select in increasing order of f_i)
      iii. Latest start time first (select in decreasing order of s_i)
      iv. Latest finish time first (select in decreasing order of f_i)
      v. Shortest activity first (select in increasing order of $f_i - s_i$)
      vi. Lowest conflict first (select the activity that has the minimum number of conflicts with the remaining tasks)

   d. Consider the recurrence $T(n) = 4T(n/2)+n$. Using Θ-notation, give a tight asymptotic bound on T(n). (E.g., T(n) is Θ(n) or Θ(n log n), etc.)

   e. What is the maximum number of edges in an undirected graph with *n* vertices, in which each vertex has degree at most *k*?

   f. You are given a connected, undirected graph G = (V, E) with positive edge weights. You form another graph G′ by squaring the weight of every edge of G. True or false: A spanning tree T is a minimum spanning tree of G if and only if T is a minimum spanning tree of G′. Explain briefly.

   g. You are given a connected, undirected graph G = (V,E) in which each edge has a numeric edge weight, and all edge weights are distinct. Let e_1, e_2, and e_3 be the edges with smallest, second smallest, and third smallest weights among all the edges of G. Among these three edges, which *must* be in the minimum spanning tree (MST) of G and which *might* be in the MST.

3. A pharmacist has W pills and n empty bottles. Let b_i denote the capacity of bottle i, that is, the number of pills it can hold. Let v_i denote the cost of purchasing bottle i. The objective is to find the least expensive combination of bottles into which to place all W pills. Describe a greedy algorithm, which given the number of pills W, the bottle capacities b_i, and the bottle costs v_i, determines the most inexpensive set of bottles needed to store all the pills. Assume that you pay only for the fraction of the bottle that is used. For example, if the i^{th} bottle is half filled with pills, you pay only $v_i/2$. (This assumption is very important.) Prove the correctness of your solution.

4. For each part, either give a short proof of the correctness of your claim (if true) or give a counterexample (if false).
    a. Consider a weighted undirected graph G. Suppose you replace the weight of every edge with its negation (e.g. w(u, v) becomes −w(u, v)), and compute the minimum spanning tree of the resulting graph using Kruskal's algorithm. True or False: The resulting tree is a maximum cost spanning tree for the original graph.
    b. Consider a weighted digraph G and source vertex s. Suppose you replace the weight of every edge with its negation and compute the shortest paths using Dijkstra's algorithm. True or False: The resulting paths are the longest (i.e., highest cost) simple paths from s to every vertex in the original digraph.

5. You are given an undirected graph G = (V, E) where each vertex is a gas station and each edge is a road with an associated weight w(u, v) indicating the distance from station u to v. The brilliant but flaky Professor X wants to drive from vertex s to vertex t. Since his car is old and may break down, he does not like to drive along long stretches of road. He wants to find the path from s to t that minimizes the maximum weight of any edge on the path. Give an O(m log n) algorithm to do this, where n = |V | and m = |E|. Briefly justify your algorithm's correctness and derive its running time.