Homework 5

Fundamental Algorithms, Fall 2004, Professor Yap

Due: Thu December 9, in class

INSTRUCTIONS:

- NOTICE: In the last homework, some students copied programs from each other. THIS IS NOT ACCEPTABLE FOR ANY PART OF YOUR HOMEWORK. While group discussions are encouraged, each submitted homework *must* represent your individual work. It must NOT be copied from anywhere, including group members. For future reference, please report the names of your discussion group members.

---

1. (10+15+10 Points) MST
   You must know how to do hand-simulations of the algorithms of Prim and Kruskal.
   (a) Please carry out the hand simulations of these two algorithms on the graph of Figure 7 (Lecture V). The edge weights are described in Exercise 5.1.
   (b) Let $S \subseteq E$ be a set of edges, and S is acyclic (i.e., S is a forest). To implement Kruskal's algorithm, we need to test whether a given edge e will create a cycle when added to S. We want you to implement this test by using a simple data structure, denoted $D(S)$: it is essentially a list of linked lists. Each linked list of $D(S)$ corresponds to the set of nodes in a tree of S. Assume that the vertex set is $V = \{1, 2, \ldots, n\}$. If $i \in V$, let $L(i)$ denote the linked list corresponding to i. Show how to implement the following queries on $D(S)$ efficiently:

   - Given $i, j \in V$, is $L(i) = L(j)$?
   - Given $i, j \in V$, merge the list of $L(i)$ and $L(j)$. After merging, we have $L(i) = L(j)$.

   NOTE: you may introduce auxilliary data structures as needed.
   (c) Implement Kruskal's algorithm using $D(S)$, and give a complexity analysis of your implementation.

2. (15+15 Points) Convex Hull
   Let $S_n = (v_1, \ldots, v_n)$ be an input sequence of points in the plane. Assume that the points are sorted by x-coordinates and satisfy $v_1 <_x v_2 <_x \cdots <_x v_n$. Note that "$a <_x b$" means that $a.x < b.x$ where a, b are points. Our goal is to compute the upper hull of S_n. In stage i ($i = 1, \ldots, n$), we have processed the sequence S_i comprising the first i points in S_n. Let H_i be the upper hull of S_i. The vertices of H_i are stored in a push-down stack data structure, D. Initially, D contain just the point v_1.
   (a) Describe a subroutine $Update(v_{i+1})$ which modifies D so that it next represents the upper hull H_{i+1} upon the addition of the new point v_{i+1}. HINT: Assume D contains the sequence of points $(u_1, \ldots, u_h)$ where $h \geq 1$ and u_1 is at the top of stack, with $u_1 >_x u_2 >_x \cdots >_x u_h$. For any point p, let $LT(p)$ denote the predicate `LeftTurn`(p, u_1, u_2). If $h = 1$, $LT(p)$ is defined to be `True`. Implement $Update(v_{i+1})$ using the predicate $LT(p)$ and the (ordinary) operations of push and pop of D.
   (b) Using part (a), describe an algorithm for computing the convex hull of a set of n points. Analyze the complexity of your algorithm.

3. (5+15 Points) Splay Analysis
   In this question, we define the potential of node u to be $\Phi(u) = \lg(\textsc{Size}(u))$, instead of $\Phi(u) = \lfloor \lg(\textsc{Size}(u)) \rfloor$.
   (a) How does this modification affect the validity of our Key Lemma about how to charge `splayStep`? HINT: In fact, we now have $\Phi'(u) - \Phi(u)$ is *strictly positive*. This appears to make our proof easier, but what could go wrong?
   (b) Cosider Case I in the proof of the Key Lemma. Show that if $\Phi'(u) - \Phi(u) \leq \lg(6/5)$ then $\Delta\Phi = \Phi'(w, v) - \Phi(u, v) \leq -\lg(6/5)$. HINT: the hypothesis implies $a + b \geq 9 + 5c + 5d$.

4. (15+10+10 Points)
   (a) Compute the length of the longest common subsequence, the edit distance and the alignment distance of the following pair of strings: $X =$ `agacgttcgtta` and $Y =$ `cgactgctgt`. These are the

---

functions $L(X,Y), D(X,Y), A(X,Y)$. You must organize your 3 computations in the form of matrices, as in the Lecture Notes.
(b) Suppose you want to compute $L(X,Y,Z)$, the length of the longest common subsequence for three strings X, Y, Z. State and prove the dynamic programming principle analogous equation (2) in Lecture VII.
(c) Describe an algorithm for $L(X,Y,Y)$ in pseudo code. Analyze its complexity as a function of $m = |X|, n = |Y|, p = |Z|$.

5. (10+20 Points) Dynamic Programming
The following exercises are found in Lecture VII.
(a) Exercises 4.2, the optimal order for multiplying matrices. (Read section 5 for more information about optimal multiplication of matrices.)
(b) Exercises 4.3, a wavelet computation problem.

6. (0 Points)
The problems below carry no credit. Please do not hand such problems; they are for your own practice. We strongly recommend that you do them.

7. (a) Show that the minimum spanning tree T of an undirected graph G with distinct weights must contain the edge of smallest weight. (b) Must it contain the edge of second smallest weight? (c) Must it contain the edge of third smallest weight?

8. Our cost model and analysis of binary counters can yield the exact cost (not just an upper bound) of incrementing from any initial counter value to any final counter value. Show that the exact cost to increment a counter from 68 to 125 is 190.

9. Perform the following splay tree operations, starting from an initially empty tree.

$$\texttt{insert}(3,2,1,6,5,4,9,8,7), \texttt{lookUp}(3), \texttt{delete}(7), \texttt{insert}(12,15,14,13), \texttt{split}(8).$$

10. Suppose we want to compute the edit distance $D(X,Y)$ in which each delete and insert operations has cost 1, but each replace operation has a cost depending a and b, where we want to replace symbol a by symbol b. Let $r(a,b)$ denote this cost. We assume $0 < r(a,b) < 2$. Design an efficient algorithm for this problem.