Exercise 4 - Solutions

December 4, 2003

Question 1

a) Let x and y be two strings. $LCE(x, y)$ denotes the minimal length of a string that contains both x and y as substrings, and $LCS(x, y)$ denotes the maximal length of a common substring of x and y.

Claim 1 *For any two strings x and y, with lengths n and m respectively, $LCE(x, y) = m + n - LCS(x, y)$.*

Proof Let $x = x_1 x_2 \cdots x_n$, and $y = y_1 y_2 \cdots y_m$. Assume $x_{i_1} \cdots x_{i_k} = y_{j_1} \cdots y_{j_k}$ is a common substring of maximal length ($k = LCS(x, y)$). Then the string

$$x_1 \cdots x_{i_1-1} y_1 \cdots y_{j_1-1} \mathbf{x_{i_1}} x_{i_1+1} \cdots x_{i_2-1} y_{j_1+1} \cdots y_{j_2-1} \mathbf{x_{i_2}} \cdots \mathbf{x_{i_k}} x_{i_k+1} \cdots x_n y_{j_k+1} \cdots y_m$$

has length $m + n - LCS(x, y)$. This shows that $LCE(x, y) \leq m + n - LCS(x, y)$. On the other hand, if w is a minimal common extension. Let I be the set of indices of w that form x and J the set of indices of w that form y. Thus, the length of w is equal to $|I| + |J| - |I \cap J|$. Since if we take w in the coordinates that belong to $I \cap J$ we get a common substring of x, y, we get that $LCE(x, y) \geq m + n - LCS(x, y)$. ■

b) We have shown in the first section that if we have a common substring of x and y of maximal length we can easily construct an extension of x and y with minimal length. Thus it is enough to know how to find a common substring with maximal length. This can be done using dynamic programming, by solving the following sub-problems: For $i = 1, \ldots, n, j = 1, \ldots, m$, let $L_{i,j} = LCS(x_1 \cdots x_i, y_1 \cdots y_j)$, and $D_{i,j}$ is 1 if $x_i = y_j$, 2 if $L_{i-1,j} > L_{i,j-1}$ and 3 otherwise. The sub-problems satisfy the following property: If $D_{i,j} = 1$ then $L_{i,j} = L_{i-1,j-1} + 1$, else $L_{i,j} = \max\{L_{i,j-1}, L_{i-1,j}\}$. After computing the values of all the sub-problems we can construct a common substring of maximal length (and also a minimal extension) by backtracking using the values of the $D_{i,j}$'s, i.e. we start with pointers at the end of x and y, and we then go backwards writing the common substring along the way. At each step, when the pointers are at x_i and y_j we do the following: If $D_{i,j} = 1$ we write x_i in the common substring, if $D_{i,j} = 2$ we go one step backwards in x (for the LCE also write x_i), otherwise we go one step backwards in y (for the LCE also write y_j). When $i = 0$ or $j = 0$ we go all the way back in y or x respectively.

Question 2

The following theorem was proved in class.

Theorem 2 *Consider a finite set S, and let $F \subset 2^S$ be a non empty hereditary family of subsets of S. The greedy algorithms works for (S, F) for every weight function if and only if F satisfys the exchange property.*

We can use this to show that the greedy algorithm for maximal matching does not work. Let $S = E$, be the set of edges in G, and denote by F the family of all matchings in G. It is not hard to see that F is hereditary (every subset of edges in a matching is also a matching). But a simple example shows that F does not satisfy the exchange property: Consider the complete graph on 4 vertices v_1, v_2, v_3, v_4. The set $\{(v_1, v_2), (v_3, v_4)\}$ is a matching, and so is the set $\{v_2, v_3\}$. But $\{(v_1, v_2), v_2, v_3)\}$ and $\{(v_2, v_3), (v_3, v_4)\}$ are both not matchings.

Question 3

Follows directly from the definitions.