CPSC-320 Intermediate Algorithm Design and Analysis Winter 2000

Practice Homework # 4

1. Cormen, Leiserson and Rivest, problem 16-3, page 325: When a "smart" terminal updates a line of text, replacing an existing "source" string $x[1 \ldots m]$ with a new target string $y[1 \ldots n]$, there are several ways in which the changes can be made. A single character of the source string can be deleted, replaced by another character, or copied to the target string; characters can be inserted; or two adjacent characters of the source string can be interchanged ("twiddled") while being copied to the target string. After all the other operations have occurred, an entire suffix of the source string can be deleted, an operation known as "drop to end of line."

   As an example, one way to transform the source string "algorithm" into the target string "altruistic" is to use the following sequence of operations.

Operation	Target string	Source string
copy a	a	lgorithm
copy l	al	gorithm
replace g by t	alt	orithm
delete o	alt	rithm
copy r	altr	ithm
insert u	altru	ithm
insert i	altrui	ithm
insert s	altruis	ithm
twiddle it into ti	altruisti	hm
insert c	altruistic	hm
drop hm	altruistic	

   There are many other sequences of operations that accomplish the same result.

   Each of the operations delete, replace, copy, insert, twiddle, and drop has an associated cost. The cost of a given sequence of transformation operations is the sum of the costs of the individual operations in the sequence. For the sequence above, the cost of converting "algorithm" to "altruistic" is

$$3 \cdot \text{cost(copy)} + \text{cost(replace)} + \text{cost(delete)} + 3 \cdot \text{cost(insert)} + \text{cost(twiddle)} + \textit{cost}(\textit{drop}).$$

   Given two sequences $x[1 \ldots m]$ and $y[1 \ldots n]$, the **edit distance** from x to y is the cost of the least expensive transformation sequence that converts x to y. Describe a dynamic programming algorithm to find the edit distance from x to y and print an optimal transformation sequence. You should assume that the costs of the transformation operations are also part of the input to your algorithm; i.e. the correctness of your algorithm should not rest on any assumptions about the relative cost of two operations. Analyze the running time and space requirements of your algorithm.

2. **Tandem arrays** A substring α contained in string S is called a *tandem array* of β (called the base) if α consists of more than one consecutive copy of β. For example, if $S = xyzab\text{-}cabcabcabcpq$ then $\alpha = abcabcabcabc$ is a tandem array of $\beta = abc$. Note that S also contains a tandem array of *abcabc*, i.e. a tandem array with a longer base. A *maximal* tandem array is a tandem array that cannot be extended either left or right. Given the base β, a tandem array of β in S can be described by two numbers (s, k) giving its starting location in S and the number of times β is repeated. A tandem array is an example of a repeated substring; identification of tandem arrays arises in analysis of genomic DNA molecules.

   (a) Suppose S has length n. Give an example to show that two maximal tandem arrays of a given base β can overlap.

   (b) Give an $O(n)$ time algorithm that takes S and β as input, finds every maximal tandem array of β, and outputs the pair (s, k) for each occurrence. (Since maximal tandem arrays of a given base can overlap, a naive algorithm would establish only an $O(n^2)$ time bound.)

3. Formulate a problem based on handing back class assignments in a room with 180 people. What are the allowable "operations"? What cost would you assign to these operations? (Try to keep things simple!)

   Given your problem formulation, design the best algorithm you can to solve the problem.