Lab4

Edit distance

In order to transform one source string of text $x[1..m]$ to a target string $y[1..n]$, we can perform various transformation operations. Our goal is, given x and y, to produce a series of transformations that change x to y. We use an array z—assumed to be large enough to hold all the characters it will need—to hold the intermediate results. Initially, z is empty, and at termination, we should have $z[j] = y[j]$ for $j = 1, 2, \ldots, n$. We maintain current indices i into x and j into z, and the operations are allowed to alter z and these indices. Initially, $i = j = 1$. We are required to examine every character in x during the transformation, which means that at the end of the sequence of transformation operations, we must have $i = m + 1$.

The are 3 transformation operations:

Replace: put a character c to $z[j]$ (c may be the same as $x[i]$), by setting $z[j] \leftarrow c$, and then incrementing both i and j.

Delete: skip a character in x by incrementing i but leaving j alone.

Insert: put the character c into z by setting $z[j] \leftarrow c$ and then incrementing j, but leaving i alone.

1. Given two sequences $x[1..m]$ and $y[1..n]$ and set of transformation-operation costs, the edit distance from x to y is the cost of the least expensive operation sequence that transform x to y. Write a dynamic-programming algorithm that finds the edit distance from $x[1..m]$ to $y[1..n]$ and prints an optimal operation sequence.

Hint:
计算出三种操作的状态转移方程，注意每种操作对于i，j的变化都是不同的。最后推导出完整的状态转移方程。

将代码写到提供的Lab4.java中

Deadline：2014.11.19　17:10:00