Problem Set 5

Noella James

03/17/2017

collaborators: none

Problem 5-2: Broken Keys

Recurrence Equation

The problem requires a algorithm to transform string1 to string2. Lets represent string1 as X and string2 as Y. X can be represent as a set of symbols $(X_1, X_2, \ldots X_m)$. Y can be represent as a set of symbols $(Y_1, Y_2, \ldots Y_n)$. The problem indicates that the number of presses required to enter a character x is $N[x]$. The problem does not specify the number of times the delete key has to be pressed to delete a character from string1 (X). We present the value as α.

In the derivation of the optimal solution there are 4 cases to be considered.

Case 1

The last character in string X is not required for string Y and thus should be deleted. The recurrence relation is
$OPT(m, n) = \alpha + OPT(m - 1, n)$

Case 2

The last character in string Y need to be entered. The recurrence relation is
$OPT(m, n) = N[Y_n] + OPT(m, n - 1)$

Case 3

The last character of X and Y are same and thus the mouse pointer can be shifted one place towards left. Thus, the recurrence relationship is
$OPT(m, n) = OPT(m - 1, n - 1)$

Case 4

The last character of X and Y are different and thus the last character in X needs to be deleted and the last character Y needs to be entered. Thus, the recurrence relationship is
$OPT(m,n) = \alpha + N[Y_n] + OPT(m-1, n-1)$

Recurrence equation

The algorithm should chose the minimum value of these 4 options. Thus, the final recurrence relationship is
$OPT(m,n) = min[\alpha + OPT(m-1,n), \quad N[Y_n] + OPT(m,n-1), \quad OPT(m-1,n-1), \quad \alpha + N[Y_n] + OPT(m-1,n-1)]$

Algorithm

Algorithm 1 MINIMUM KEYSTROKES

procedure MIN-KEYSTROKES(X, Y) ▷ $X = \{X_1, X_2, \ldots, X_m\}, Y = \{Y_1, Y_2, \ldots, Y_n\}$, Array $N[x]$ = number of times a character x is pressed for it to be typed, α = number of times delete key has to be pressed to delete a character
  Array $A[0 \ldots m, 0 \ldots n]$
  Initialize $A[i, 0] = i\alpha$ for each i
  for $j = 1 \ldots n$ **do**
    $A[0, j] = A[0, j-1] + N[x_j]$
  end for
  for $j = 1 \ldots n$ **do**
    for $i = 1 \ldots m$ **do**
      $A[i,j] = min[\alpha + A(i-1,j), \quad N[y_j] + A(i,j-1), \quad A(i-1,j-1), \quad \alpha + N[y_j] + A(i-1,j-1)]$ ▷ Recurrence relationship as seen above
    end for
  end for
  return $A[m, n]$
end procedure

Complexity

String1 = X = $\{X_1, X_2, \ldots, X_m\}$ has m characters. String2 = Y = $\{Y_1, Y_2, \ldots, Y_n\}$ has n characters. The outer for loop runs n times and the inner loop runs m times. The complexity of the recurrence statement is $O(1)$. Therefore, the total complexity is $O(nm)$.

Correctness

Lemma 0.1. *The algorithm $MINIMUMKEYSTROKES(X, Y)$ correctly computes the minimum number of keystrokes needed to convert X to Y.*

Proof. We will prove this by induction on $i + j$.

Base Case:

When $i + j = 0$, we have $i = j = 0$, and we don't have to convert any strings. The function returns 0, which is correct since 0 keystrokes are needed to convert X to Y. Indeed $OPT(0, 0) = 0$.

Induction Hypothesis:

Now consider arbitrary values of i and j, and suppose the statement is true for all pairs (i', j') with $i' + j' < i + j$. The four cases that exist are
1. We delete X_i.
2. We type Y_j.
3. X_i is the same as Y_j and nothing needs to be done.
4. We delete X_i and type Y_j.
We choose the minimum of the four options. Thus,

$$\begin{aligned} A[i,j] &= min[\alpha + A(i-1,j), \quad N[y_j] + A(i,j-1), \quad A(i-1,j-1), \quad \alpha + N[y_j] + A(i-1,j-1)] \\ &= min[\alpha + OPT(i-1,j), N[y_j] + OPT(i,j-1), OPT(i-1,j-1), \alpha + N[y_j] + OPT(i-1,j-1)] \\ &= OPT(i,j) \end{aligned}$$

Thus, this completes the proof. ∎