Homework Assignment 1

Matthieu Boyer

3 octobre 2023

Table des matières

1 Exercise 1 - [Edit Distance/Levenshtein Distance] 1
1.1 Question 1 1
1.2 Question 2 1

1 Exercise 1 - [Edit Distance/Levenshtein Distance]

1.1 Question 1

Proposition 1.1.1 (Complexity and Correction)**.** *If we denote C_f the complexity of f, this algorithm has time complexity $\mathcal{O}\left(C_f \left(\frac{n}{t}\right)^2\right)$. This algorithm is correct.*

Démonstration. — Moreover, it is clear this algorithm is correct as it only just applies the dynamic programming algorithm for the Levenshtein distance by steps.

— This algorithm complexity comes from the fact it has two while loops for which the commands are executed at most $\lceil n/t \rceil$ times. The commands in both *while* loops are executed in $\mathcal{O}\left(C_f\right)$. The *for* loops inside the *left while* loop are equivalent to loops for i between left and left $+ t - 1$ and thus are disjoint. The sum of their complexity over the *left while* loop is then n. The number of operations inside the *up while* loop is then in $\mathcal{O}\left(C_f \frac{n}{t}\right)$ and thus the total complexity is, as announced, in $\mathcal{O}\left(C_f \left(\frac{n}{t}\right)^2\right)$

■

1.2 Question 2

By the recurrence formula : $D[i][j] = \max \begin{cases} D[i-1][j] + 1 \\ D[i][j-1] + 1 \\ D[i-1][j-1] + 1 if S[i] \neq T[j] else 0 \end{cases}$ we see that $D[i][j]$ is at most 1 plus one of its left neighbour, upper neighbour or upper left corner neighbour.

Algorithme 1 Question 1 - Levenshtein Distance with f

Input S, T, f, t ▷ Two Strings, the function f computing the values and the step t

$\mathbf{D} = \text{zeros}(n+1, n+1)$ ▷ $len(S) = len(T) = n$
for $i \leftarrow 0$to $n+1$ **do**
  $\mathbf{D}[i][0] \leftarrow i$
end for

for $j \leftarrow 0$ to $n+1$ **do**
  $\mathbf{D}[0][j] \leftarrow j$
end for

$\text{up}, \text{left} \leftarrow 0, 0$
while $\text{up} < n$ **do**
  $\text{left} \leftarrow 0$
  while $\text{left} < n$ **do**
    $d \leftarrow \min(n - \text{up}, t)$
    $e \leftarrow \min(n - \text{left}, t)$

    $b \leftarrow \mathbf{D}[\text{up}][\text{left}]$
    $a \leftarrow \mathbf{D}[\text{up}+1 \rightarrow \text{up}+1+d][\text{left}]$
    $c \leftarrow \mathbf{D}[\text{up}][\text{left}+1 \rightarrow \text{left}+1+e]$

    $f(a, b, c, d, e)$ ▷ We can suppose here that f modifies only the last line and column of F in D with side-effect.
    $\text{left} \leftarrow \text{left} + e$
    for $i \leftarrow 1$ to $t-1$ **do**

$$\mathbf{D}[\text{up}+i][\text{left}] \leftarrow \min \begin{cases} \mathbf{D}[\text{up}+i][\text{left}-1]+1 \\ \mathbf{D}[\text{up}+i-1][\text{left}]+1 \\ \mathbf{D}[\text{up}+i-1][\text{left}-1]+\mathbb{1}_{\{S[up+i]=T[left]\}} \end{cases}$$

▷ We update the first Column of the block we consider.
    end for
  end while
  $\text{up} \leftarrow \text{up} + d$
  for $i \leftarrow 1$ to n **do**

$$\mathbf{D}[\text{up}][i] \leftarrow \min \begin{cases} \mathbf{D}[\text{up}][i-1]+1 \\ \mathbf{D}[\text{up}-1][+i]+1 \\ \mathbf{D}[\text{up}-1][i-1]+\mathbb{1}_{\{S[up+i]=T[left]\}} \end{cases}$$

▷ We update the first line of the blocks we will consider.
  end for
end while
return $\mathbf{D}[n][n]$