

Assignment 3 Theoretical Analysis

Liam Tolkkinen

Professor Prasad

04/11/24 (with 2-day extension: 04/13/24)

String Matching

The algorithm discussed in class for string matching utilizes a cost matrix and reverse matrix traversal. The way that the costs of the matrix are calculated is by filling in the first row and the first column with the gap cost, as predefined by the corresponding input file. There are 3 different costs to consider in this implementation:

The first is the gap penalty denoted by δ .

The second is the mismatch penalty denoted by α .

The third is the match penalty which is usually 0 or sometimes a negative penalty.

The method of traversal is to start at the matrix index (1,1) and iterate through the entire matrix. This is done to fill in the cells with the current shortest penalty to reach that cell. This is done by looking at the 3 adjacent cells and filling it in with the current α penalty (if mismatch) + the minimum penalty of the adjacent cells.

TIMING ANALYSIS

The time analysis goes as follows:

M = length of the first string

N = length of the second string

Initialization of the first column and row: $O(M + N)$.

Double for loop starting at (1,1):

Initialize the values using the minimum values:

$(O(M*N) - O(M + N)) * O(1) <--$ subtract the time to fill in the 1st row/col

Traversal to align the strings: (Average case: $O(\sqrt{M^2 + N^2})$) $<--$ because it traverses diagonally on the average case, worst case = $O(M + N)$

Combine complexities:

Final time = $O(M*N)$

SPACE COMPLEXITY:

2 strings: $O(M) + O(N)$

2D matrix for costs: $O(M*N)$

Final strings: $O(M) + O(N)$

Final space: $O(M*N)$