

# Project 2 Report

Kenneth Sills, Kevin Orr, Elijah Malaby

July 13, 2017

## 1 Breaking Down Problems

## 2 Parameters for the recursion

The two arrays  $p$  and  $t$  (pattern and typo) and indexes  $i$  and  $j$  representing the positions in  $p$  and  $t$  currently being compared

## 3 What recurrence can you use

## 4 What are the base cases

Whenever  $i$  or  $j$  are equal to 1. If  $i = 1$ , the remaining characters in  $t[1..j - 1]$  were trivially all insertions at the beginning. If  $j = 1$ , the remaining characters in  $i[1..i - 1]$  were deleted. If both  $i = 1$  and  $j = 1$ , there are no further characters to compare.

## 5 What data structure would you use

A map from pairs of  $(i, j)$  to the cost of the recurrence for  $(i, j)$ .

## 6 Pseudocode for a memoized dynamic programming solution

## 7 Worst case time complexity

$O(mn)$

## 8 Pseudocode for nested loop

## 9 Can the space complexity of the iterative algorithm be improved relative to the memoized algorithm

## 10 Describe one advantage and disadvantage of the iterative algorithm