

## CS302 - Final Review

1. **Dynamic Programming** Given two strings  $x = x_1x_2x_2 \dots x_n$  and  $y = y_1y_2y_2 \dots y_m$ , the edit distance  $D(x, y)$  is the minimum number of insertions or deletions or substitutions (a substitution involves replacing one character in the string with another) required to turn  $x$  into  $y$  (or vice versa). This is used for spellcheckers: if someone types a word that is not in the dictionary, you want to find the word that is closest to it in edit distance. For example, if someone typed “graffe” its edit distance from “graft” is 2 (delete “e”, substitute “f” for “t”), while its edit distance to “giraffe” is 1 (insert “i”).

We want to design a dynamic programming algorithm to determine the shortest edit distance.

- (a) If we consider the optimal way of turning  $x$  into  $y$ , what options do we have for how to make that transition? (Think about the e.g. smallest/first/last choices we could make.)
  - (b) Let  $\vec{x}_i$  be the first  $i$  letters in the string  $x$  and  $\vec{y}_i$  be the first  $i$  letters in the string  $y$ . Explain how  $D(x, y) = D(\vec{x}_n, \vec{y}_m)$  is related to the edit distance of substrings in each option from the previous part.
  - (c) Combine the results into one recurrence relation. What is the base case?
  - (d) Let  $A$  be the array that you creat in your algorithm. What should you store in the array? What should the size of the array be?
  - (e) Which rows should you fill out first?
  - (f) How long will it take to fill in the rest of  $A$ .
  - (g) Write pseudocode that, if given a properly filled out  $A$  as input, tells you how to turn  $x$  into  $y$ .
  - (h) Prove using a loop invariant that your pseudocode is correct. (You may assume that  $A$  is properly filled out, and that you previously proved the correctness of your recurrence relation.)
2. **NP** A variant of Subset-Sum involves a set that contains both positive and negative integers, and the target value is always 0. Call this variant Subset-Sum-0. Prove that Subset-Sum-0 is NP-Complete
  3. **Randomized Algorithms** What is the average runtime of randomized search without replacement if there are  $c$  copies of the item we are looking for out of an array of size  $n$ . Please use indicator random variables and go through the usual routine. You do not need to simplify your final answer - it can be a messy sum.