CS 331: Algorithms and Complexity
Homework II

Trung Dang Nathan Mardanov Kevin Tian

Due date: October 1, 2025, end of day (11:59 PM), uploaded to Canvas.

Late policy: 15% off if submitted late, and 15% off for every further 24 hours before submission.

Please list all collaborators on the first page of your solutions.

When runtimes are unspecified, slower runtimes than the intended solution receive partial credit.

1 Problem 1

Let X, Y be strings with lengths m and n respectively, with $m \leq n$.

(i) **(8 points)** Give an algorithm that, on inputs X, Y, returns **True** or **False** depending on if X is a subsequence of Y. For example, it should return **True** if $X =$ "arm", $Y =$ "algorithm".

(ii) **(10 points)** Give an algorithm that, on inputs X, Y, counts the number of times X appears as a subsequence of Y. For example, it should return 3 if $X =$ "an", $Y =$ "banana". You may assume for this part that adding two arbitrary integers takes $O(1)$ time.

(iii) **(2 points)** Give a runtime bound on your algorithm in Problem 2(ii) if d-digit integer addition takes $O(d)$ time. You may use the fact that $\log \binom{n}{m} = O(m \log(n))$.

2 Problem 2

RNA serves a key role in regulating your cells: for example, it makes up ribosomes, transcribes and translates genetic code into amino acids, and controls gene expression. The structure of RNA molecules is important in determining how they function.

One description of RNA molecules is their "primary structure," the string $S \in \{\text{`A'}, \text{`U'}, \text{`C'}, \text{`G'}\}^n$ of n bases that chain together to form the molecule. However, RNA also has a "secondary structure" describing how bases interact (e.g., twist and fold in on themselves). Primary structures that have a larger number of valid secondary structures are considered more likely to appear in nature.

(20 points) Design an algorithm that takes as input a primary structure and returns the number of valid secondary structures. Formally, a secondary structure is a partial matching of S to itself, where some pairs of bases in S are matched, and all other bases are left unmatched. For the secondary structure to be valid, it must additionally follow the following rules.

- *Complementary pairs.* 'A' bases may only be matched with a 'U' bases, and similarly 'C' bases may only be matched with 'G' bases.
- *No overlaps.* Let (i, j) and (k, ℓ) be two matched base pairs with $i < j$ and $k < \ell$. Then we must have $i < j < k < \ell$ or $k < \ell < i < j$ (i.e., (i, j) and (k, ℓ) are non-overlapping).
- *Sufficient slack.* No matched base pairs (i, j) can have $|i - j| \leq 4$.

For example, an RNA molecule with primary structure "AUGAGUGCAAC" has at least one valid secondary structure that matches only the indices $(3, 9)$ (although it has others as well).

3 Problem 3

Give algorithms for the following problems taking strings as inputs. For full credit, three of your algorithms should run in time $O(n^2)$, and one of your algorithms should run in time $O(n)$.

(i) **(5 points)** Return the longest common substring of length-n inputs X and Y.

(ii) **(5 points)** Return the longest palindromic subsequence of length-n input X.

(iii) **(5 points)** Return the length $n - 5$ substring of length-n input X that is smallest in lexicographical order, i.e., how the English dictionary is sorted (assume that $n > 5$, and that all of the characters in X are the usual 26 English letters).

(iv) **(5 points)** We say a string Z is a shuffle of X, Y if its characters can be partitioned into two subsets, so that deleting one subset and concatenating the other in order gives X or Y. For example, "greedy" is a shuffle of "ged" and "rey." Given length-n inputs X and Y, and a length-$2n$ input Z, return **True** or **False** depending on if Z is a shuffle of X and Y.

4 Problem 4

UT Austin and Texas A&M are having an algorithms competition. Each school sends their n strongest algorithms students, each of whom has an algorithms rating. Suppose that the UT students have ratings $\{a_i\}_{i\in[n]} \subset \mathbb{N}$, and the A&M students have ratings $\{b_i\}_{i\in[n]} \subset \mathbb{N}$.

The coaches are trying to ensure a fair competition between the two teams, so they want to pair up competitors using a permutation $\pi : [n] \to [n]$ so that the UT student with rating a_i is paired with the A&M student with rating $b_{\pi(i)}$. In particular, they want to find the permutation minimizing

$$\sum_{i\in[n]} \left(a_i - b_{\pi(i)}\right)^2, \tag{1}$$

the total squared rating deviation summed over all n assigned pairs.

(20 points) Give an algorithm which takes as input $\{a_i\}_{i\in[n]}$, $\{b_i\}_{i\in[n]}$ (as Array instances) and outputs the index pairs $\{(i, \pi(i))\}_{i\in[n]}$ given by the permutation π minimizing the expression (1).

5 Problem 5

(**20 points**) Complete the assignment at this link. This link is only accessible on your UT email.