# CS331: Algorithms and Complexity
# Homework II

Trung Dang    Ryan Park    Kevin Tian

**Due date: October 2, 2024, 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

(i) **(8 points)** Suppose you have $n$ units of a material, arranged in a rod of length $n$. You can cut the rod into pieces of integer length to sell. Give an algorithm that takes as input $P$, a length-$n$ $\mathsf{Array}$, such that $P[i] \in \mathbb{R}_{\geq 0}$ is the profit obtained by selling a length-$i$ piece of the rod for all $i \in [n]$, and returns the maximum profit achievable from cutting the rod.

(ii) **(10 points)** Suppose you have $mn$ units of a material, arranged as a $m \times n$ grid of squares. You can divide the grid into subrectangles of integer length to sell, but each time you cut a rectangle, you must divide it exactly into two subrectangles, i.e., the only cuts you can make are end-to-end vertical or horizontal cuts. Give an algorithm that takes as input $P$, a two-dimensional $\mathsf{Array}$ with dimensions $m \times n$, such that $P[i][j]$ is the profit obtained by selling a $i \times j$ subrectangle, and returns the maximum profit achievable from cutting the grid.

(iii) **(2 points)** Explain how to generalize your algorithm from Problem 1(ii) to three dimensions.

## 2 Problem 2

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 $\log \binom{n}{m} = O(m \log(n))$.

## 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 longest substring of length-$n$ input $X$ using only one unique character.

(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

(**20 points**) Let $L$ be an Array containing $n$ real numbers. Give an algorithm that, on input $L$, returns $P$, a permutation[1] of $L$ that minimizes the following (among all permutations of $L$):

$$\sum_{i \in [n]} \left( \max_{j \in [i]} P[j] - \min_{j \in [i]} P[j] \right).$$

# 5 Problem 5

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

[1] A permutation of an Array $L$ is another Array that contains all the same objects as $L$ in a possibly different order. For example, {'c', 'b', 'e', 'a', 'd'} is a permutation of {'a', 'b', 'c', 'd', 'e'}.