CSE Ph.D. Qualifying Exam, Fall 2025
Algorithms

Instructions:

Please answer three of the following four questions. All questions are graded on a scale of 10. If you answer all four, all answers will be graded and the three lowest scores will be used in computing your total. **This exam is closed-book, closed-notes.**

Questions:

1. **Greedy: Video Streaming**

   A video streaming service sends data from a central server to a user in fixed-size data packets of maximum size W bytes. The video is split into chunks, and each chunk i has a size of w_i bytes. Chunks must be sent in the order they appear in the video stream.

   Due to network constraints, only one packet can be prepared at a time. The current transmission policy is greedy: the system fills the packet with as many chunks as will fit, in order, and as soon as the next chunk doesn't fit, it sends the packet and starts a new one.

   Some engineers wonder whether this greedy approach uses more packets than necessary. Perhaps, in some cases, sending a partially full packet earlier could allow the following packets to be more tightly packed overall.

   Prove that, for a given sequence of video chunks and fixed packet size W, the greedy strategy actually minimizes the number of packets sent.

   Hint: establish the optimality of this greedy algorithm by identifying a measure under which it "stays ahead" of all other solutions.

2. **Dynamic Programming: Cable Assembly**

   You are setting up a network in an office and need to run a cable that is exactly *target_length* meters long. You have access to pre-cut cables of fixed lengths—for example, 1m, 3m, and 5m—and you can buy as many of each type as you need.

   You're given a list *cable_lengths*, where each element represents the length of a cable segment available for purchase. Your goal is to use as few cable segments as possible, such that the total length is exactly *target_length* meters.

   Write pseudocode of a dynamic programming algorithm that returns the fewest number of cables that you need. If it's not possible to reach the exact length using the available cable segments, return -1.

   Example: *cable_lengths* = [1, 2, 5], *target_length*=11

   Solution: 3 (5 + 5 + 1)

3. **Dynamic Programming: Neat Printing**

Consider the problem of neatly printing a paragraph with a monospaced font (all characters having the same width). The input text is a sequence of n words of lengths $\ell_1, \ell_2, \ldots, \ell_n$, measured in characters, which are to be printed neatly on a number of lines that hold a maximum of M characters each. No word exceeds the line length, so that $\ell_i \leq M$ for $i = 1, 2, \ldots, n$. The criterion of "neatness" is as follows. If a given line contains words i through j, where $i \leq j$, and exactly one space appears between words, then the number of extra space characters at the end of the line is $m - j + i - \sum_{k=i}^{j} \ell_k$, which must be nonnegative so that the words fit on the line. The goal is to minimize the sum, over all lines except the last, of the cubes of the numbers of extra space characters at the ends of lines.

Give a dynamic-programming algorithm that runs in $O(n^2)$ time and space to print a paragraph of n words neatly. Show that your algorithm achieves the desired bounds.

4. **NP-complete: 2-Partition**

The **2-Partition Problem** is defined as follows:

> **Input:** A set of n positive integers $A = \{a_1, a_2, \ldots, a_n\}$.
> **Question:** Can A be partitioned into two disjoint subsets A_1 and A_2 such that
>
> $$\sum_{a_i \in A_1} a_i = \sum_{a_j \in A_2} a_j?$$

Prove that the 2-Partition problem is NP-complete using the fact that the **Subset Sum** problem is NP-complete.

The Subset Sum problem is defined as follows:

> **Input:** A set $S = \{x_1, x_2, \ldots, x_m\}$ of m positive integers and a target integer T.
> **Question:** Does there exist a subset $S' \subseteq S$ such that
>
> $$\sum_{x_i \in S'} x_i = T?$$

Remember to include all steps of NP-completeness proof.