CS 630, Fall 2024, Homework 2
Due Wednesday, September 25, 2024, 11:59 pm EST, via Gradescope

Homework Guidelines

Collaboration policy Collaboration on homework problems, with the exception of programming assignments and reading quizzes, is permitted, but not encouraged. If you choose to collaborate on some problems, you are allowed to discuss each problem with at most 5 other students currently enrolled in the class. Before working with others on a problem, you should think about it yourself for at least 45 minutes. Finding answers to problems on the Web or from other outside sources (including generative AI tools or anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, you should write "Collaborators: none." It is a violation of this policy to submit a problem solution that you cannot orally explain to an instructor or TA.

Typesetting Solutions should be typed and submitted as a PDF file on Gradescope. You may use any program you like to type your solutions. LaTeX, or "Latex", is commonly used for technical writing (`overleaf.com` is a free web-based platform for writing in Latex) since it handles math very well. Word, Google Docs, Markdown or other software are also fine.

Solution guidelines For problems that require you to provide an algorithm, you must provide:

1. pseudocode and, if helpful, a precise description of the algorithm in English. As always, pseudocode should include
   - A clear description of the inputs and outputs
   - Any assumptions you are making about the input (format, for example)
   - Instructions that are clear enough that a classmate who hasn't thought about the problem yet would understand how to turn them into working code. Inputs and outputs of any subroutines should be clear, data structures should be explained, etc.

   If the algorithm is not clear enough for graders to understand easily, it may not be graded.
2. a proof of correctness
3. an analysis of running time and space .

You may use algorithms from class as subroutines. You may also use facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand.

Problem 1 Price contest with a blackbox algorithm (10 points)

You are a contestant on a TV show, and Congratulations, you won! As your price you get to choose what to take home. Specifically, there are n objects on display, each with a retail value $[a_1, a_2, \ldots, a_n]$, you are also given a budget B. You can choose any subset of the items as long as their total value is within budget. Your goal is to pick a subset of total value exactly B.

In this problem we do NOT ask you to solve this problem from scratch [1]. Instead, we want you to solve it through a polynomial-reduction style approach. Specifically, suppose that you have access to a blackbox algorithm $CanSolve(\ )$. The algorithm takes as input a list of integers $[x_1, x_2, \ldots, x_k]$ and a target value T. The output is $CanSolve([x_1, x_2, \ldots, x_k], T) = True$ if and only if there is a subset of the x_i values, such that there sum is exactly T. Otherwise it returns $False$.

Design an algorithm, that using $CanSolve(\ )$ returns the set of items that you should select, if there is a subset with total value exactly B, otherwise return "not possible". In your running time analysis use $O(C)$ to represent the runtime of $CanSolve(\ )$. The algorithm should take $[a_1, a_2, \ldots, a_n]$ and B as input.

Problem 2 Vaccine testing (10 points)

You are developing a new vaccine and are ready for human trials! For efficiency, you want to keep your trial as small as possible. Each person has some health-characteristics, e.g. age, weight, disease history, etc. Your goal is to select a subset of individuals to receive vaccines, such that each person's set of characteristics are well represented in your trial.

For privacy reasons you don't have access to the actual medical data of the population. Instead, you are given for every two persons p_i and p_j their distance $d(p_i, p_j)$. Distances are symmetric. (You don't have to worry about how exactly the distances are computed.) You may assume that the data is given to you in the input format of a 2D-table D, such that $D[i][j] = d(p_i, p_j)$.

Given a target value T, we say that a set of trial participants is *representative* of the population if for each person p there is a trial participant with distance at most T from p.

In this problem you have access to a blackbox algorithm called $SCSolver(\ )$. This algorithm solves an instance of the minimum Set Cover problem; given a universe $U = \{u_1, u_2, \ldots, u_n\}$ of n items and given m subsets $S = \{S_1, S_2, \ldots S_m\}$ of U, it returns a minimum number of sets in S, such that each item in U is covered. (*Meaning that S returns the actual sets that form a minimum set cover.*)

Design an algorithm for the Vaccine Testing problem that makes calls to $SCSolver(\ )$. The input to your algorithm is the distance table D and the max distance value T. The output should be a minimum representative set of individuals. In your runtime analysis you may assume that $SCSolver(\ )$ takes $O(C)$ time.

[1] You will NOT receive credit for any algorithm that doesn't use CanSolve( ) in a meaningful way.