CSE Ph.D. Qualifying Exam, Spring 2026
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: Mars Habitat**

   A space colony consists of n habitat modules, all of which must be supplied with life support. There is no central life-support hub.

   Each module i can be supplied with life support in one of the following ways:

   (a) Install a local life-support unit inside module i at cost `unit`$[i]$.

   (b) Establish a life-support link from module i to another module j that is already supplied with life support, at a cost determined by the connection between modules i and j.

   You are given:

   - An array `unit`, where `unit`$[i]$ is the cost of installing a local life-support unit in module i.
   - A list `links`, where each tuple (i, j, c) indicates that a life-support link between modules i and j can be established at cost c.

   Your goal is to ensure that *all modules are supplied with life support* at *minimum total cost.*

   (a) Describe a polynomial-time greedy algorithm to solve this problem.

   (b) Prove that your algorithm is correct. Your proof may use an exchange argument.

2. **Dynamic Programming: Task Selection**

   A robot is preparing for a mission and has a limited battery capacity W.

   There are n available tasks. Each task i:

   - consumes w_i units of battery, and
   - yields a mission reward v_i.

   Each task can be executed at most once, and the total battery consumption of the selected tasks must not exceed W.

   Write pseudocode of a dynamic programming algorithm that determines which tasks the robot should perform in order to maximize total mission reward without exceeding the battery capacity. State the running time of your algorithm.

3. **Dynamic Programming: Corporation Tree**

   Professor Stewart is consulting for the president of a corporation that is planning a company party. The company has a hierarchical structure; that is, the supervisor relation forms a tree rooted at the president. The personnel office has ranked each employee with a conviviality rating, which is a real number. In order to make the party fun for all attendees, the president does not want both an employee and his or her immediate supervisor to attend.

   Professor Stewart is given the tree that describes the structure of the corporation. Each node of the tree holds, in addition to the pointers, the name of an employee and that employee's conviviality ranking.

   Given this corporation tree with n nodes, design an algorithm to make up a guest list that maximizes the sum of the conviviality ratings of the guests. Your algorithm should run in $O(n)$ time. Show that your algorithm runs in the desired time.

4. **NP-complete: Exam Design**

   At the end of the semester, a professor must design the final exam for her class. She has a collection of potential exam problems, and for each problem she knows which students would really enjoy that problem.

   Formally, the input consists of:

   - A set of students $U = \{s_1, s_2, \ldots, s_m\}$.
   - A set of exam problems $P = \{p_1, p_2, \ldots, p_n\}$.
   - For each problem $p_j \in P$, a subset $L(p_j) \subseteq U$ indicating the students who enjoy problem p_j.
   - An integer k.

   The professor would like to select a subset of problems $P' \subseteq P$ such that:

   (a) $|P'| \leq k$, and

   (b) For every student $s \in U$, there exists at least one problem $p \in P'$ with $s \in L(p)$.

   The EXAMDESIGN problem is to answer the question: does such a subset P' exist?

   Prove that the EXAMDESIGN problem is NP-complete, given that the VERTEXCOVER problem is NP-complete. VERTEXCOVER(G, k) is the problem of deciding whether in graph G there exists a subset of vertices S of size at most k such that all edges in G have at least one endpoint in the selected vertex set S.