Practice Problems

Dynamic Programming

a) Billboards on a Highway:

Suppose you are managing the construction of billboards on a Highway, a heavily travelled stretch of road that runs west-east for M miles. The possible sites for billboards are given by numbers X= x1, x2, . . . , xn, each in the interval [0,M] (specifying their position along the highway, measured in miles from its western end). If you place a billboard at location xi, you receive a revenue of ri> 0. Regulations imposed by the county's Highway Department require that no two of the billboards be within less than or equal to 5 miles of each other. You'd like to place billboards at a subset of the sites to maximize your total revenue, subject to this restriction. Device an efficient algorithm that takes the input X and compute the sites for billboards that maximize the revenue. Give the pseudocode of your algorithm, also compute its time complexity.

Greedy Algorithms

a) Minimize Waiting Time:

A server has n customers waiting to be served. The service time required by each customer is known in advance: it is t_i minutes for customer i. So if, for example, the customers are served in order of increasing i, then the i th customer has to wait $\sum_1^i t_i$ minutes. We wish to minimize the total waiting time $T = \sum_{i=1}^{n}$ (Time spent waiting by cutomer i).

Give an efficient algorithm for computing the optimal order in which to process the customers. Also compute its time complexity.

b) Activity Selection Problem:

Part 1:

Suppose that instead of always selecting the first activity to finish, we select the last activity to start that is compatible with all previously selected activities. Describe how this approach is a greedy algorithm, and prove (see theorem 16.1) that it yields an optimal solution.

Write pseudocode (both recursive as well as iterative) to solve the activity selection problem using the above-mentioned greedy approach.

Part 2:

Not just any greedy approach to the activity-selection problem produces a maximum-size set of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from among those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the

fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

Part 3:

Consider a modification to the activity-selection problem in which each activity a_i has, in addition to a start and finish time, a value v_i . The objective is no longer to maximize the number of activities scheduled, but instead to maximize the total value of the activities scheduled. That is, we wish to choose a set A of compatible activities such that $\sum_{a_k \in A} v_k$ is maximized. Give a **polynomial-time** algorithm for this problem.
