15-451/651 Algorithm Design & Analysis, Spring 2023

Recitation #5

Objectives

- Practice dynamic programming algorithms for sequences and trees (using small steps).
- Understand how to analyze the complexity of a dynamic programming algorithm.

Recitation Problems

1. **(Removal Game)** You are playing a game with Wurzelbrunft. There is a list of n numbers L, and players move alternately (with you going first). For each move, the player either removes the first element or the last element from the list, and adds that value to their score. Both of you are trying to maximize your score. Come up with an algorithm to find what your score would be, assuming both of you play optimally.

   This can be solved in $O(n^2)$ using Dynamic Programming.

   (a) Find an appropriate sub-problem.

   (b) Given a sub-problem, come up with a recurrence to determine its answer.

   (c) Given this, finish the solution by describing how to compute the answer for the whole array.

2. **(Tidying Up)** You are in a room containing n objects at integer coordinates (x_i, y_i) for $1 \leq i \leq n$. In the center of the room at $(0, 0)$, you have a box in which you would like to place all of the objects. The problem is that the objects are quite heavy, so you can only manage to carry at most *three* of them at a time! The box is also heavy, so you can not move the box. The time that it takes you to move between two points i and j is $|x_i - x_j| + |y_i - y_j|$.

   Assuming that you start at position $(0, 0)$ with the empty box, what is the minimum amount of time required for you to place all of the objects into the box? Give an algorithm for this problem that runs in $O(2^n \cdot n^2)$ time.

   Hints:

   - Any solution to this problem will essentially be a bunch of trips away from the box to fetch items, and then back to the box to put them away. Does the relative ordering of which trip happens first matter? [No.]

3. **(Cheapest Tree Separation)** There are N cities, numbered from 1 through N, connected by $N-1$ roads, forming a weighted tree. Countries A and B each occupy a set of cities (no city is occupied by both countries, and some cities may not be occupied at all).

   To stop fighting between the two countries, you want to destroy roads such that no city occupied by country A is connected to a city in country B. Destroying a road of length x costs x dollars. What is the minimum cost required? Given a linear time algorithm.