ADA 2021 Tutorial 5

Jatin, Syamantak

1. Define the subproblems clearly
2. Write a recursion using the above definition
3. Argue properly about why the recursion is correct (this is the optimal substructure property)
4. Implement using tables and argue runtime.

1 Jatin turns DJ Again.

(From Midsem ADA 2021) Fed up with the accidental question paper leak, Jatin decides to turn DJ once again. Now, he has n club nights he can possibly perform at, for simplicity the ith club night is on night i. If he performs on night i, he earns money v_i and this costs him energy w_i. Now, every day he eats food (regardless of whether he has a performance that night) which gives him exactly 1 unit of energy. Jatin starts with an initial energy of 0, so just before the first night, his energy is exactly 1 (from the food that day).

Jatin needs to make sure that his energy never goes below zero else he won't be able to teach ADA ever again. In other words, if Jatin performs at a subset S of nights then S should satisfy

$$i - \sum_{j \in S : j \leq i} w_j \geq 0 \text{ for all } 1 \leq i \leq n$$

Give a polynomial-time (in n) algorithm to find the subset of club nights Jatin can perform at to make the most money, while ensuring that his energy always stays non-negative. Assume that w_i, v_i are integers.

Solution.

2 Inventory Management

Consider the following inventory problem. You are running a store that sells some large product (let us assume you sell trucks), and predictions tell you the quantity of sales to expect over the next n months. Let d_i denote the number of sales you expect in month i. We will assume that all sales happen at the beginning of the month, and trucks that are not sold are stored until the beginning of the next month. You can store at most S trucks, and it costs C to store a single truck for a month. You receive shipments of trucks by placing orders for them, and there is a fixed ordering fee of K each time you place an order (regardless of the number of trucks you order). You start out with no trucks. The problem is to design an algorithm that decides how to place orders so that you satisfy all the demands d_i, and minimize the costs. In summary: There are two parts to the cost.

- First, storage: it costs C for every truck on hand that is not needed that month. Second, ordering fees: it costs K for every order placed.
- Each month, you need enough trucks to satisfy the demand d_i, but the amount left over after satisfying the demand for the month should not exceed the inventory limit S.

Solution.

3 Knapsack cover

You are given items i with sizes w_i and values v_i, and a target value V. Find a subset of items of minimum total size with total value at least V (or decide that there is no such subset).

Solution.