CS 161 (Stanford, Winter 2024) Lecture 14

Pre-lecture exercises will not be collected for credit. However, you will get more out of each lecture if you do them, and they will be referenced during lecture. We recommend writing out your answers to pre-lecture exercises before class. Pre-lecture exercises usually should not take you more than 30 minutes.

Pre-Lecture Exercises

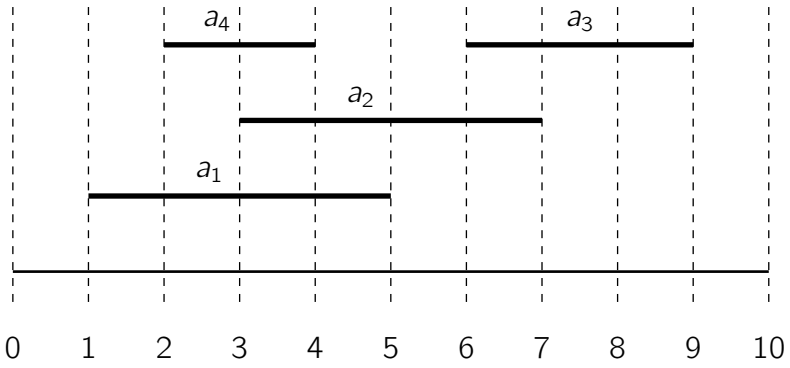
In this pre-lecture exercise you'll consider **greedy** algorithms for two problems: **unbounded knapsack** and **scheduling**. Here are the two problems:

Problem 1: Unbounded knapsack. You have access to an unlimited supply of items $a_1, \ldots, a_n$. Item a_i has weight w_i and value v_i. You want to fill a knapsack of weight W with items (repetitions are allowed) so that (a) the sum of the items in your knapsack is at most W and (b) the value of the items in your knapsack is as large as possible.

(We saw this problem in Lecture 13, along with a DP solution).

Problem 2: Activity selection. There are activities $a_1, \ldots, a_n$ that you would like to do. Activity i starts at time s_i and finishes at time f_i. In order to do activity i, you will be occupied for the whole range $[s_i, f_i]$, and cannot do any other activity. Out of this set of tasks, you'd like to find out the maximum number of activities possible.

For example, if there were four tasks, with $(s_1, f_1) = (1, 5)$, $(s_2, f_2) = (3, 7)$, $(s_3, f_3) = (6, 9)$ and $(s_4, f_4) = (2, 4)$, then the picture would be:



and the optimal solution would be two activities. You could either do a_1 and a_3, or a_4 and a_3.

1. We saw a DP solution for **problem 1** in class last week. Your friend suggests a different algorithm. They reason: "The best thing to do is to take the item that has the biggest value/weight ratio. Since repetitions are allowed, just take as many copies of this best item as you can fit in your knapsack."

   Do you think your friend's reasoning is correct?

2. Your friend has a similar sort of reasoning for **problem 2**. They argue: "The best thing to do is to choose the activity (that's not blocked by previous activity choices) with the smallest finishing time. That way, you get to do an activity and leave as much room as possible for future activities." That is, your friend suggests that you first choose the activity with the smallest finishing time, then next choose the activity that's not blocked with the smallest finishing time, and so on until you can't choose activities anymore.

   Do you think your friend's reasoning is correct?