Homework 6

Due: Friday, Oct 17, 11:59 PM PT

For each of the questions below, design a dynamic programming based algorithm by answering the following sub-parts:

a) Define in plain English the **subproblems** that your dynamic programming algorithm solves. (In other words, your answer should be something like "Opt(x,y,z..) denotes the min/max/?? value obtained from x items/people/?? , with y time/money/?? , z __" and so on. Do not merely paraphrase the given problem or describe the high-level approach etc.) (4 points)

b) Define a recurrence relation that expresses the value of each subproblem in terms of the smaller subproblems. (6 points)

c) Using the recurrence formula in part b, write pseudocode to find the solution.
Make sure you specify all the base cases and their values, as well as the final answer to the problem. (6 points)

d) What is the time complexity of the solution? Further, determine whether it is an efficient solution or not. (2+2 points)

1. James wants to surprise his fiancée with exquisite jewellery made from gold. He owns many gold mines and wants the goldsmith to make different items. He gives the goldsmith a list of n different types of items, where each item type weighs w_i grams and each has a value of v_i. James' fiancée likes jewellery but only has space for some items, as her jewellery box has a maximum capacity of W grams. James wants the goldsmith to make items from the list and insists that the combined weight of all the items must be equal to or less than W grams, but at the same time, maximizing the total value of all the created items. Design a DP algorithm to output the total optimal value of the items, whose total weight doesn't exceed W grams. Note that the final jewellery box can contain multiple instances of the same jewellery. (20 points)

2. You are participating in a game show, where you are presented with a line of briefcases. On top of each briefcase, you can see the amount of money you can win by choosing that briefcase. However, after each time you pick out a briefcase, the host will remove the two briefcases adjacent to it (or the only adjacent briefcase if the corner-one was picked). Your goal is to maximize the amount of money you can win; however, if your final amount collected is more than K dollars, you will lose all your money. Design a dynamic

programming algorithm to compute the maximum earnings possible without exceeding K dollars by picking suitcases as per the aforementioned rules.

For example, if your starting set of briefcase values is [44, 51, 46, 55, 23, 12] and you want to make at most 100$. Here is how your selection can play out:
You pick the 44$ briefcase. The host removes the 51$ briefcase, and the remaining array would be [46, 55, 23, 12]. You now pick the 55$ briefcase and the host removes the 46$ and 23$ briefcases, and the remaining array would be [12]. You stop playing since you don't want to make more than 100$, therefore the game ends, and 99$ is the maximum amount you can make playing this game. (20 points)

3. Suppose you have a rod of length N, and you want to cut up the rod and sell the pieces in a way that maximizes the total amount of money you get. A piece of length i is worth p_i dollars. Devise a Dynamic Programming algorithm to determine the maximum amount of money you can get by cutting the rod strategically and selling the cut pieces. (20 points)

Ungraded Problems:

1. Given a set of numbers, design a DP algorithm to check whether it can be partitioned into two subsets such that the sum of the elements in both subsets is equal. (20 points)

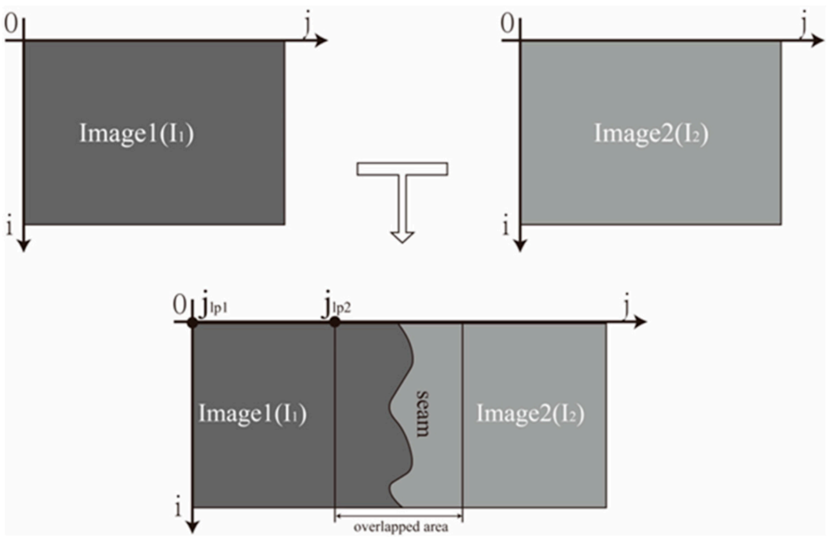
2. You are planning your holiday and have a list of events you'd like to attend. Each event is described by $[s_i, e_i, v_i]$, where s_i and e_i are the start day and end day of the event, and v_i represents how important you think the event is. Consider that:
    1. You can only attend one event at a time.
    2. If you choose to take one event, you must attend the entire event.
    3. You can attend k events at most.

   Design a DP algorithm that maximizes the total value of the events you attend. (20 points)

3. Shawn decided to go on a trip for spring break, so he decided to visit San Francisco and went to see his favorite monument, the Golden Gate Bridge. Immediately, Shawn decided to take a picture of the magnificent bridge; however, due to its size, he was unable to fit the entire bridge in one picture. Relying on his expertise as a computer scientist, Shawn decided to take two pictures from the bridge and stitch the images together when he got back home. Now Shawn is back from his trip and can't wait to show his friends the pictures from his trip. He knows that simply putting his two images together would show the lack of his photographic skills to his friends, so he decides to stitch the two images

together such that the stitching is not obvious. To do this, he knows that the two images have the same height H and share 5 columns of pixels together, so he calculates the absolute difference between the pixels of the two images in the shared area. His goal is to find a path through the pixels in the shared area that minimizes the total sum of pixel distances. Furthermore, he knows that once he selects a pixel, he can select one that is either directly above it or diagonally above it as the next pixel, because otherwise the stitching becomes way too obvious. Design a DP algorithm to help Shawn find the best path to stitch his images so that he can show off in front of his friends. (20 points)

The figure below provides an abstract representation of what Shawn is attempting to do.



4. Solve Kleinberg and Tardos, Chapter 6, Exercise 5. (20 points)