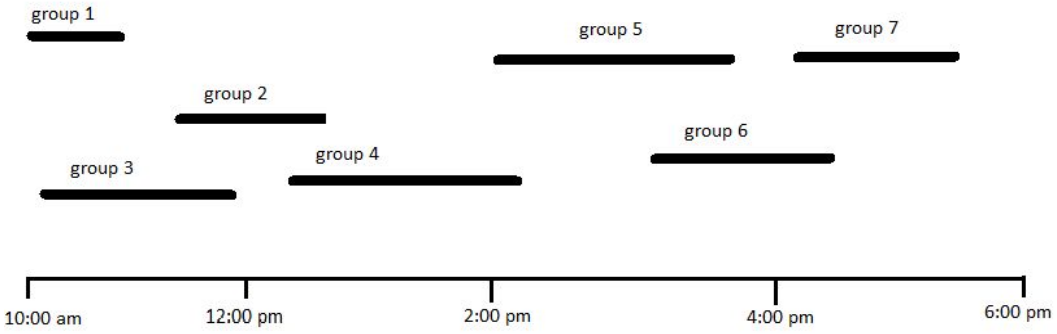
CSE221: Algorithms
Worksheet 8
Greedy Algorithms and Dynamic Programming

1. What do you understand by greedy algorithms?How will you determine if a problem can be solved using a greedy algorithm?
2.
    a. I have a party venue that I rent out for 30000 TK for weddings and birthday parties. I have received booking requests from 7 groups for arranging parties on next Friday. The requested booking time for each group is shown in the figure below. I want to maximize my earning on Friday by renting the room out to maximum number of groups. Run an Interval Scheduling algorithm for me on Friday's schedules and show me the events I can book for maximum income. Show or explain clearly how the algorithm is working.

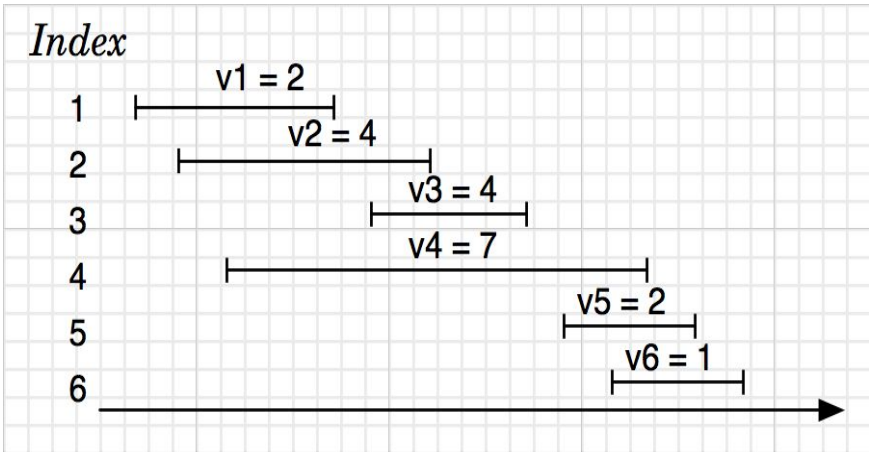


    b. If I change my pricing policy and charge more for wedding parties than birthday parties (e.g. 50000 for weddings and 20000 for birthdays), then can I use the same algorithm as above? Explain your answer with example [Assume the odd numbered groups want to book for a birthday party and the even no. groups have requested to book for a wedding]. **N.B. You may also use your own example instead of the above figure.**
    c. I have expanded my event venue recently and now I have several rooms to book several events at once. Schedule all the booking requests in the above figure using minimum number of rooms. State the total number of rooms required and show all steps for assigning each event clearly.

3.

a. A Weighted Interval Scheduling problem is defined as follows: given a set of weighted intervals, the goal is to find the optimal subset, where the intervals are non-conflicting and the total weight is maximized. The recursive equation for calculating the optimal solution for the subproblems in case of a dynamic programming solution is: OPT(j) = max(wj + OPT(p(j)),OPT(j − 1)). **Explain the meaning and reasoning of the equation clearly.** [wj = weight of interval j, j= intervals from 1 to n, n=total number of intervals, p(j) is the last interval that does not conflict with interval j].

b. Apply a suitable algorithm to the weighted intervals in figure below to find the optimal schedules. Show all steps clearly.



4.

a. You need to carry a few items in your knapsack which can take a maximum weight of 10 kg. You are allowed to take fractional items (minimum weight you can choose is 1 kg). You primary goal is to maximize the value of items in your knapsack without exceeding the maximum weight. The items, their weight and price is given below. Apply a suitable algorithm and show the quantity of each item selected and the total benefit or value.

   i. Board Markers - 2 Kg - 300 Tk
   ii. Note books - 4 Kg - 800 Tk
   iii. Glitter Pens - 2 Kg - 250 Tk
   iv. Paint brushes -3 Kg - 900 Tk
   v. Acrylic Paint - 6 Kg -3000 Tk

b. If you are not allowed to carry fractional item(i.e. You either pack the whole item or you don't take it) then will the algorithm in 3a) work? Explain with example.

c. Use a suitable algorithm to select the items and calculate the maximum benefit if fractional items are not allowed. [Hint: 0/1 knapsack]

d. The recursive equation for calculating the optimal solution for the subproblems in case of a dynamic programming solution for a 0/1 Knapsack problem is given as:

$$Opt(j,\ w)\ =$$

$$Opt(j-1,\ w)\ if\ w_j\ >\ w\ else\ Max(V_j + Opt(j-1,\ w-w_j), Opt(j-1,\ w))$$

Explain the meaning and reasoning of the equation clearly [w_j = weight of each item j, w=total weight]

5.

a. Given the following coin denominations, C= {10, 6, 1}, apply a suitable algorithm to find the minimum number of coins required to change amount, A = 13

b. Why can't the above problem be solved using Greedy algorithm. Explain with example.

c. Explain the meaning of the following recursion equation:

$$OPT(p) = \begin{cases} 0 & \text{if } p = 0 \\ min_{i:c_i<p}\{1 + OPT(p - c_i)\} & \text{if } p > 0 \end{cases}$$

6. Find the longest common subsequence for the following strings:

X= "human"
Y= "chimpanzee"

Show the cost matrix and direction matrix. What is the length of the LCS, how can it be found in the cost matrix? Explain how you will find the LCS from the direction matrix?

7. How will you determine if a problem can be solved using a dynamic programming? What kind of problems should be solved using memoization?