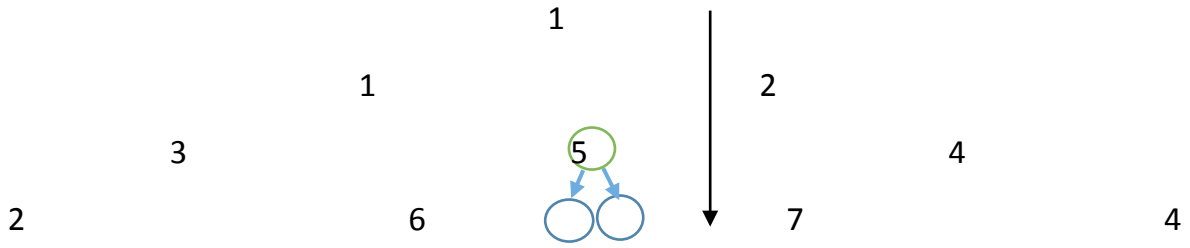
Project 2

Provide an efficient dynamic programming solution in all questions and analyze the complexity of your solution.

You are required to solve all questions with dynamic programming in this project. Provide an iterative implementation similar to the examples we have done in class. Do not use graph algorithms to solve the questions.

QUESTION 1

In this question, you are going to play a game in which your goal is to maximize the total score while you are trying to reach from top of the mountain to the bottom. As it is shown in the figure below, there is only one constraint about the movement.



The arrow shows the direction of the desired movement. The starting point is "1" in the top row and the final point is one of the numbers in the last row that is ["2", "6", "7", "4"]. The constraint about the movement is that from any point you can only go to one of the adjacent points in the next row as shown in the example above. In this example, from "5" in the third row, the possible points that you can go are "6" or "7" in the fourth row. In this question, the input is list of lists and the output will be the best route from top to bottom and the maximum total score. If there are k rows in the mountain, there

will be k lists in your input list. Lengths of these lists are 1,2,3,…,k.

Input:

mountain=*[[1], [1, 2], [3, 5, 4], [2, 6, 7, 4]]*

Output:

Route: (0, 0), (1, 1), (2, 1), (3, 2)

Score: 15

Name your function as ***'bestRoute(mountain)'***

QUESTION 2

You are operating a venue that has only **one conference hall** and you want to schedule the conferences in a given day to maximize the daily profit. For each possible conference, you are given a start time, an end time and the number of participants. You are paid a fixed amount per participant. Since you have only one conference hall, the challenge is the selection of a combination of conferences to maximize the total number of participants during this day; but, of course, you have to guarantee that there is no time conflict among the selected conferences. For example:

Conference 1	Start time: 13:00	End time: 15:59	Participants: 300
Conference 2	Start time: 11:00	End time: 13:59	Participants: 500
Conference 3	Start time: 16:00	End time: 17:59	Participants: 200

There are two possible combinations without any time clash: (Conference 1, Conference 3) and (Conference 2, Conference 3). The total number of participants are 500 and 700, respectively, so, the best choice is (Conference2, Conference 3). There is no limitation on the number of conferences that can be held in one day. Only constraint is time conflict. Input in this question is a dictionary where the key is a conference name and the value is a list that has the start time, the end time and the number of participants in this order and the output will be the names of the chosen conferences (keys in the dictionary) and the total number of participants. For example:

Input:

conferences= {"Conference 1": [1300, 1559, 300],

"Conference 2": [1100, 1359, 500],

"Conference 3": [1600, 1759, 200]}

Output:

Selected conferences: (Conference 2, Conference 3)

Total number of participants: 700

Name your function as ***'bestSelection(conferences)'***

QUESTION 3

In this question, there will be given an positive integer and you will be required to find the possible ways to write this integer as the sum of two or more positive integers. For example, the input is 4, there are 4 possible ways as listed below so the output will be 4.

3+1

2+2

2+1+1

1+1+1+1

Make sure that (3+1) and (1+3) (similarly, the combinations of other options) are considered same.

The input will be the integer and the output will be the possible ways as explained above.

Name your function as ***'possibleCombinations(n)'***