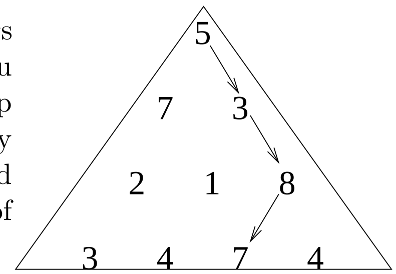
# Homework Assignment #4
Due: Apr 3, 2019, 10pm on D2L

---

[40] 1. **Moving on a pyramid**

Your input is a list of $n$ rows of numbers, where $i^{th}$ row has $i$ numbers in it. You can visualize them as arranged in a triangular shape. You start from the top number (that is, number 1 in row 1) and in one step you can reach one of the two number in the row below, either directly to the right or to the left of the current number. Your goal is to find the path from the top number to the bottom row such that the sum of numbers in your path is the largest.




Assume that your algorithm gets as an input a $n \times n$ array $C$, where $C(i,j)$ holds the value of $j^{th}$ number in row $i$. The values $C(i,j)$ for $j > i$ are undefined ($-\infty$).

(a) Define a suitable array $A$ to store partial solutions to the problem and state how to find from it the sum of the numbers on an optimal path. Hint: use a two-dimensional array.

(b) Give a recurrence to compute elements of $A$, including initialization. Justify your answer.

(c) Show how your algorithm works (i.e., fill the array) for the above example.

(d) Give pseudocode for a dynamic programming algorithm that finds the *sum of numbers* in the optimal path.

(e) Explain (in text and/or pseudocode) how to recover the actual optimal path from from the array you computed. You can add information to the arrays when computing them, if you wish.

[40] 2. **Dispensaries arrangement**

After legalization of cannabis, a certain town announces rules for locations of cannabis shops: they would like to have them all located on the main business street, however they do not want to allow dispensaries in any consecutive blocks of houses, or two in the same block. A group of enterpreneurs wanting to open dispensaries sits down with a list of available properties to decide where to open shops. For each property, they have a estimate how much profit it might bring.

Here, you will use dynamic programming to solve this problem. More specifically, you are given a list of potential shops, one per block on the business street, with a profit estimate for that shop, that is, an array of addresses and profits $((a_1, p_1), \ldots, (a_n, p_n))$. Now you need to decide at which of these locations to open shops to maximize the profits, while satisfying the town's requirement of no two shops in consecutive blocks (the other requirements are already taken care of; you can also assume that properies are sorted by increasing street number: $a_1 < a_2 < \cdots < a_n$). For example, if the input is $((105, 12), (270, 10), (340, 15), (450, 30), (533, 10))$, then the best solution would be to open shops at addresses 105 and 450, for the total profit of 42.

(a) Define array(s) you can use to solve the this problem. That is, say what the dimension(s) of your array(s) are, what is the content of the cells, and how to find the best profit from fully computed array(s). (Hint: use two one-dimentional arrays, $A_{in}$ and $A_{out}$).

(b) Give a recurrence to compute elements of your arrays, including initialization.

(c) Show the filled arrays for the above example produced by your algorithm.

(d) Give pseudocode for a dynamic programming algorithm that finds the *total profit* of the most profitable combination of shops.

(e) Explain (in text and/or pseudocode) how to recover the actual list of addresses of shops in the best arrangement from the arrays you computed. You can add information to the arrays when computing them, if you wish.

[20] 3. **Offering courses**

In a certain department at MUN, there are $n$ instructors and $m$ different courses that the department can offer. For every instructor $i$, there is a list of courses $C_i \subseteq \{1, \ldots, m\}$ that this instructor can teach. In one semester, an instructor can teach no more than three courses.

(a) Given the $n, m$ and $C_i$ for $1 \leq i \leq n$, determine the maximal number of different courses the college can offer in one semester. Hint: design a flow network that represents this problem.

(b) Give an example of an input ($n, m$ and all $C_i$) and the corresponding network where the college cannot offer all courses, although there is somebody who can teach each course.

(c) How can you recover the resulting assignment of instructors to courses given the maximum flow on your flow network?