# Homework 12

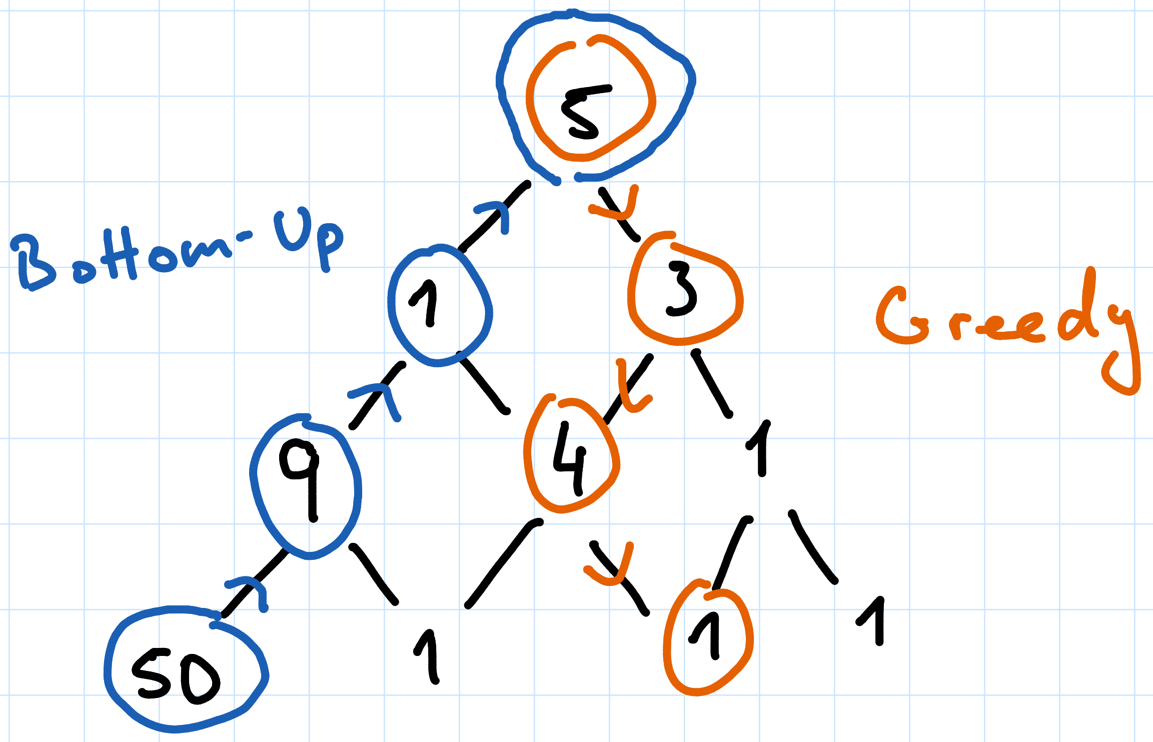
## Problem 12.2

b) I used a bottom-up approach, where I store the max path from the bottom. The construction of the table is $\Theta(n)$ since I am only going through each element once. At each element I compute the max-path leading to it (this is $\Theta(1)$ since the max-path is calculated for each of the element's children.

Hence, I am left with a recursion tree with $\Theta(1)$ operations for each element. Assuming $n$ elements, this gives us the time complexity of $\underline{\underline{\Theta(n)}}$.

A brute-force approach would calculate the sum of each path. The number of paths to the end would be $2^{k-1}$, where $k$ is the height of the triangle. At a height $k$, we will have $k$ elements at that height, hence the total number of elements will be $\sum_{i=1}^{k} i$, which is $O(n^2)$ elements. Therefore, the time complexity of this approach would be greater than the bottom up approach, since we are dealing with exponential time complexity vs linear.

c) Greedy would not always work because it would select the highest child without taking into consideration the sub-children, leading to cases such as:


Bottom-Up
Greedy
5
1
3
9
4
1
50
1
1
1