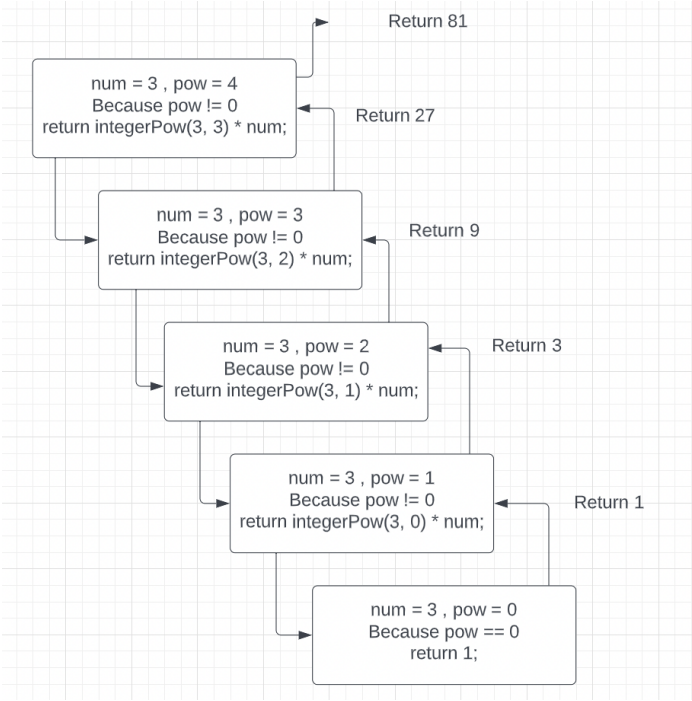
3.1:

1. The base case is when pow==0, because 2^0 is one and the most basic operation is x(x(x(x))).

2. My reduction operation is to decrement the pow variable down to pow==0, thus creating a pow number of recursive calls.

3.



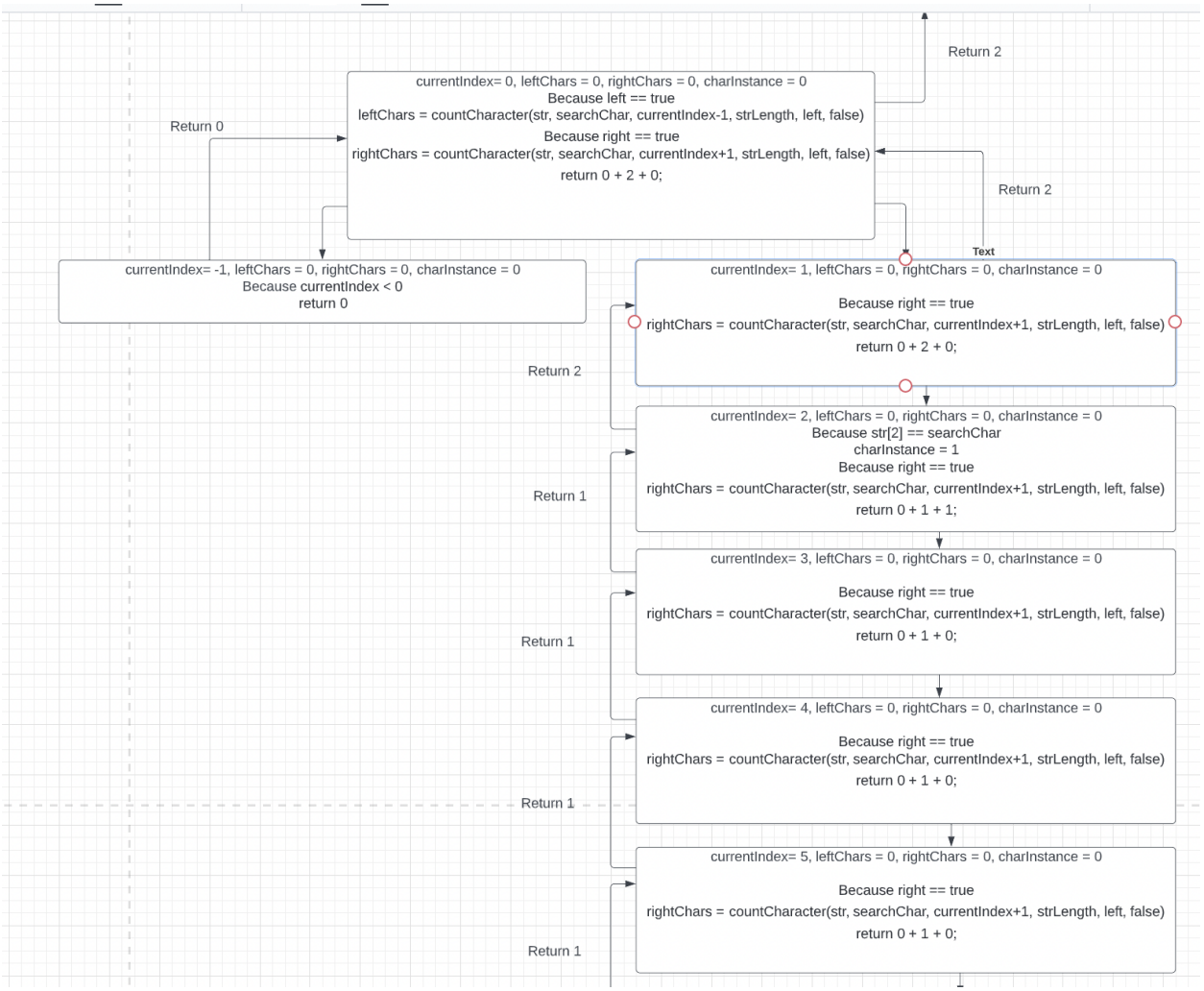
4. This is a root tree.

3.2:

1. There are four base cases, the first is when the current index is beyond the bounds of the array. The second is when the current index contains the search char. The third is when there are elements to the left of the index, and the fourth is when there are elements to the right of the array.

2. The reduction operations are when we increase and decrease the current index along the array in both directions, element by element until it reaches the ends of the array.

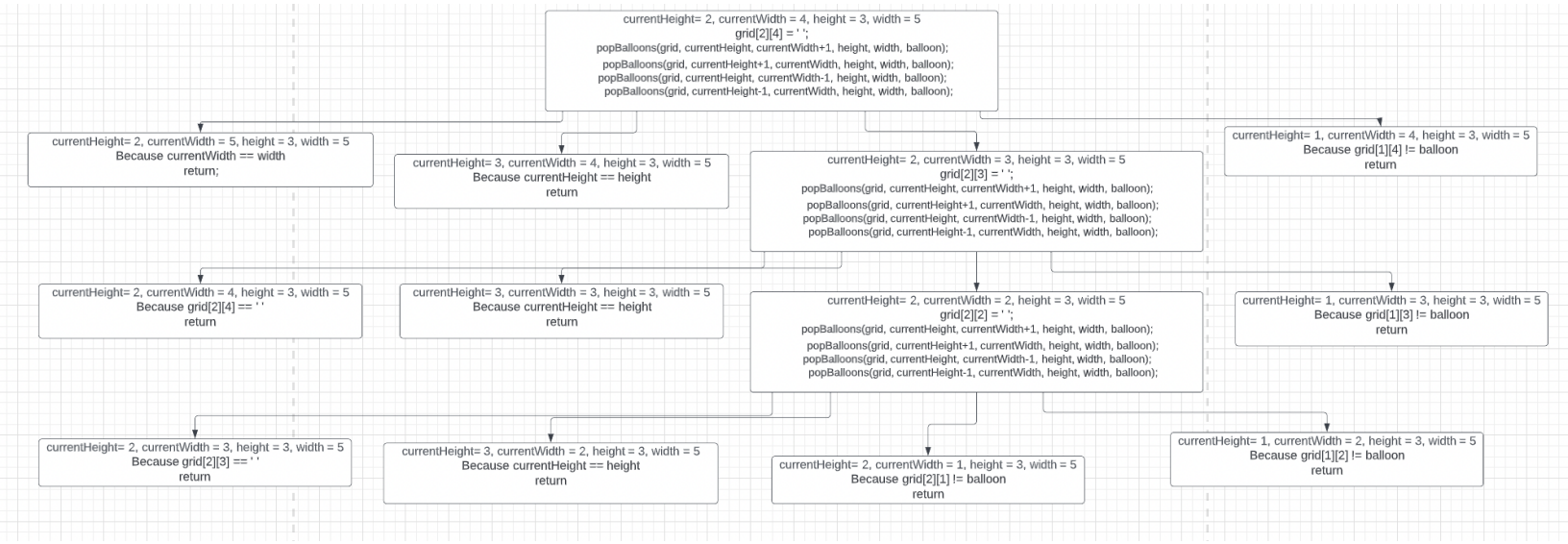
3. Look down



4. It makes a binary tree (somewhat a root tree as well)

3.3:

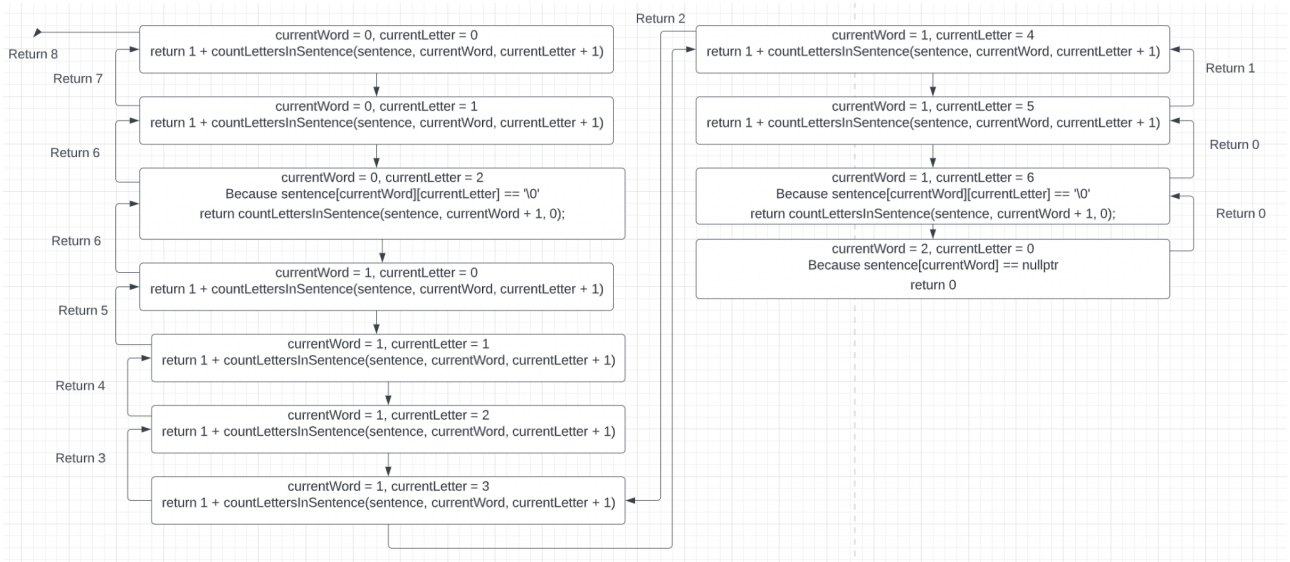
1. There are two base cases, one: if the current height or width is out of bounds, exit. Two: if the coordinates do not contain a balloon or they contain an empty space, exit, otherwise place a space.
2. The reduction operations are checking the up, down, right, left, connecting coordinates for a balloon, until it reaches the end of the grid.
3.



4. This makes a general tree (4 child nodes or none)

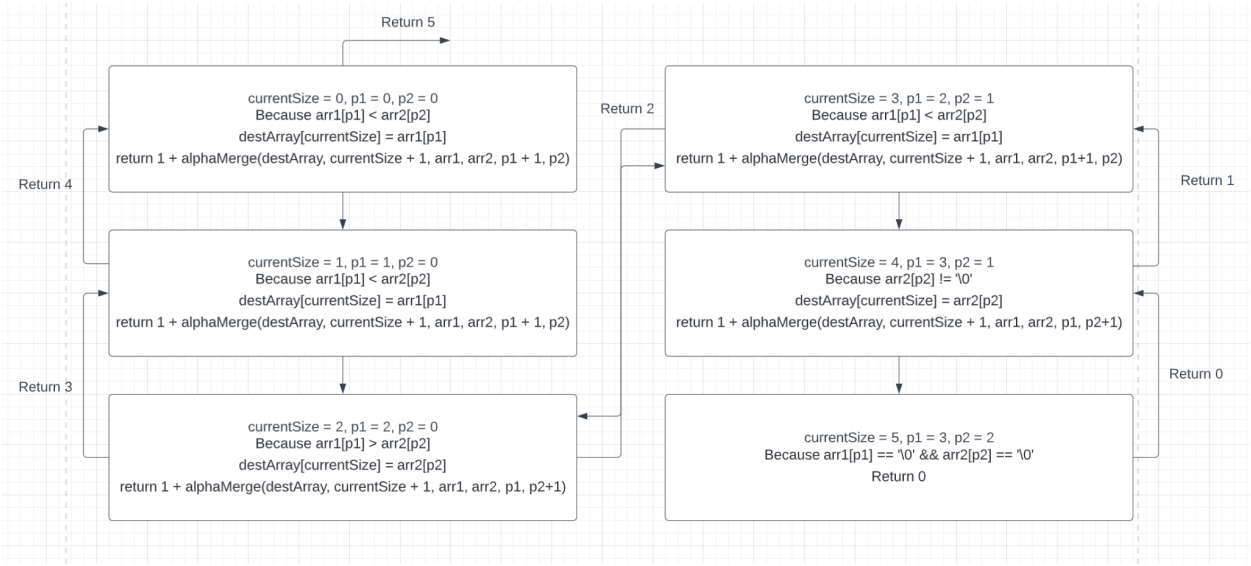
3.4:

1. There are two base cases, one: if the current word is nullptr (the last word) in which the function terminates. Two: if the current letter is '\0', in which the function calls the next word.
2. The reduction operations are incrementing the current word till the word hits null, as well as incrementing the current letter until it hits null.
3.



4. This is a root tree.

3.5:
1. There are four base cases, one: if both arrays are done, exit. Two: if both aren't done, compare them and place the smaller one in the combined array, then increment that arrays index. For three and four, if only one of the arrays are done, place each element of the other array in the combined array until its done.
2. The reduction operations are incrementing the combined array, comparing the two arrays at their given indexes, and then placing it into the combined array while incrementing the smaller array's index.
3.



4. This is a root tree.