

**Cankat Adiloğlu**

21426515

Computer Science

## ALGORITHM

Here i explain an algorithm that has been changed since the design report, and more specific.

1. Draw an  $N \times N$  matrix that all components are 0.
2. Randomly make one of those 0's 1. If it's on the top line, directly make it 2.
3. When a cell made 1, check if there is any 2 near it. If there is, change 1 to 2.
4. When a cell made 2, check all 4 sides of it. If there is any 1, change it to 2.
5. Repeat step 4 until there is no 1's near 2's.
6. Check if water reaches the bottom line. If yes, end this round. If not, repeat step 2.

## FUNCTIONS

**void form\_matrix(int size, int \*\*ground);**

Allocating a region of memory by "calloc" that can hold the values for matrix's items.

"size" is the size of the one side of the matrix (which is expected from the user to identify).

"ground" is an integer array that it's one element represents a line for matrix. I used a "for loop" to allocate memor for both lines and columns.

**void flow(int \*\*ground , int line , int column);**

"flow" is a recursive function that makes water (2) flow to the open cells next to it (1).

Function checks if water can flow up first. If it flows, program does the same thing for the new watered cell. After it finishes flowing up, does the same for down, left and right.

"ground" is a double-dimension integer array that i used for the matrix.

"line" is the element of the array's first dimension. In other words, the line of the cell we are processing.

"column" is the element of the array's seconds dimension. In other words, the column of the cell we are processing.

**void move\_number(int \*\*ground, int size , float \*count , int move);**

At the end of the program, we count the 1's and 2's by using "move\_number" function.

So the number of the non-zero elements are the number of moves we did to let water reach bottom line.

**"count"** is an integer array and it's elements defined as 0 first. At the end of each round, "count[move]" becomes the number of non-zero elements in the matrix. So it returns the number of changes made to let water reach bottom.

**"move"** is an integer which represents the number of move (T). It increments by 1 at the end of every move(round).