# Problem 3

The input is an N by N matrix of nonnegative integers that is already in memory. Each individual row is increasing from left to right. Each individual column is increasing from top to bottom. Propose an O(N) worst-case algorithm that decides if a number x is in the matrix.

We need to do a binary search on a two dimentinal array. The first search will search the colum for the row that has an int that is closest to the number x while still being <= the number x (the number you are searching for).

1 2 3 13 x = 13

6 7 8 14

**11 12 13 15**

15 16 18 30

Then we can test each index in the row if it contains the number. If there is an exact match, we can return true.

1 2 3 13 x = 13

6 7 8 14

11 12 13 15 OK @ [2][2]

15 16 18 30

If there is no exact match, decrement the chosen column index by one and search that row

1 2 3 13 x = 13

6 7 8 14

11 12 14 15 x

15 16 18 30

Previous Row

1 2 3 13 x = 13

6 7 8 14 x

11 12 13 15

15 16 18 30

Previous Row

1 2 3 13 OK x = 13

6 7 8 14

11 12 13 15

15 16 18 30

If not found in the final row, return false