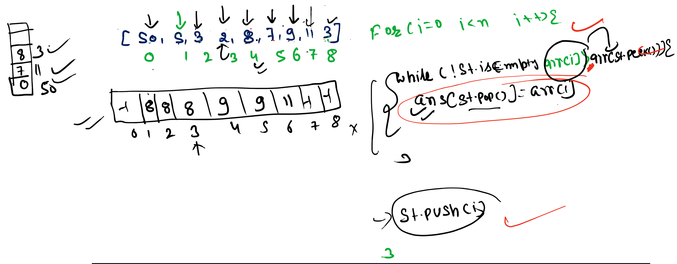
Lec-28

16/06/24 12:52 PM

Given an array, print the Next Greater Element (NGE) for every element. The Next Greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.



The stock span problem is a financial problem where we have a series of N daily price quotes for a stock and we need to calculate span of stock's price for all N days. You are given an array of length N, where i^{th} element of array denotes the price of a stock on i^{th}. Find the span of stock's price on i^{th} day, for every 1<=i<=N.

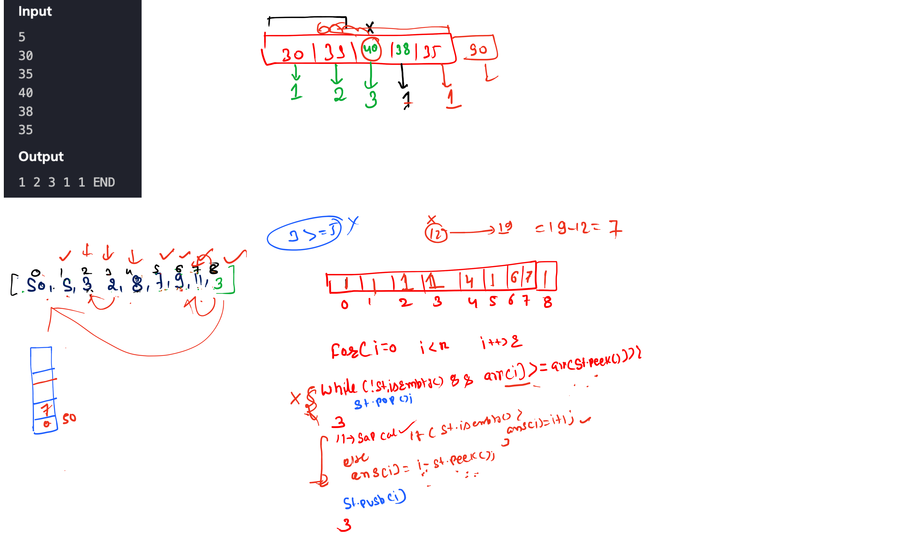
A span of a stock's price on a given day, i, is the maximum number of consecutive days before the $(i+1)^{th}$ day, for which stock's price on these days is less than or equal to that on the i^{th} day.

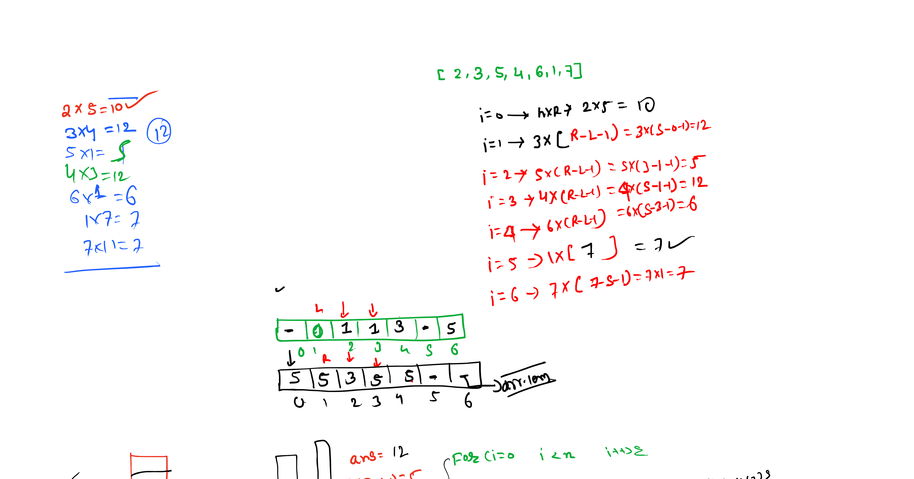
Input

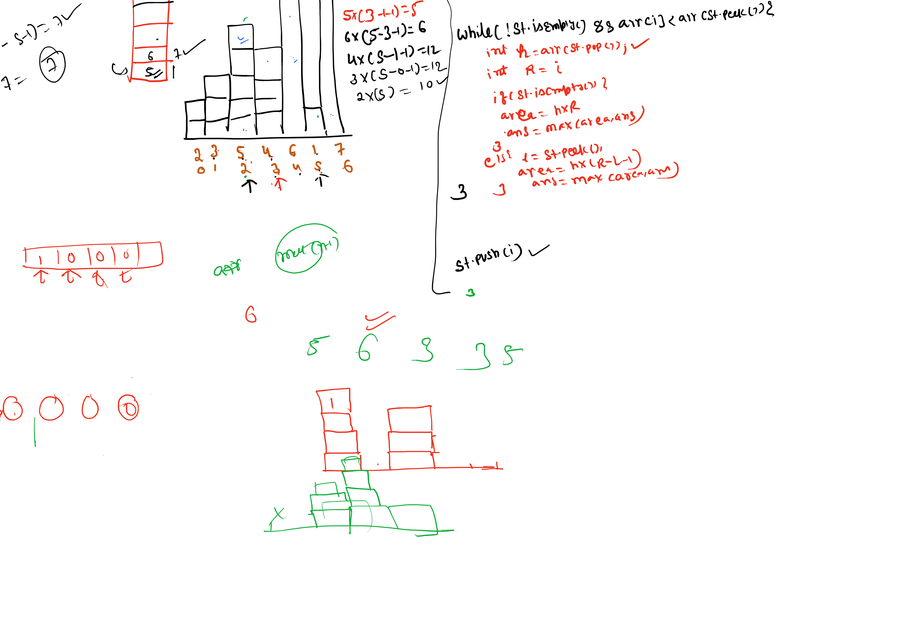
```
5
30
35
40
38
35
```

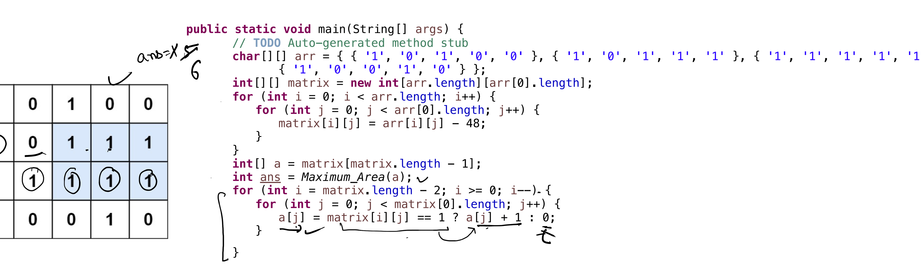
Output

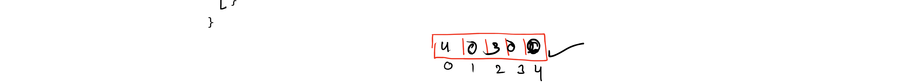
```
1 2 3 1 1 END
```











```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    char[][] arr = { { '1', '0', '1', '0', '0' }, { '1', '0', '1', '1', '1' }, { '1', '1', '1', '1', '1' },
            { '1', '0', '0', '1', '0' } };
    int[][] matrix = new int[arr.length][arr[0].length];
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[0].length; j++) {
            matrix[i][j] = arr[i][j] - 48;
        }
    }
    int[] a = matrix[matrix.length - 1];
    int ans = Maximum_Area(a);
    for (int i = matrix.length - 2; i >= 0; i--) {
        for (int j = 0; j < matrix[0].length; j++) {
            a[j] = matrix[i][j] == 1 ? a[j] + 1 : 0;
        }
    }
}
```