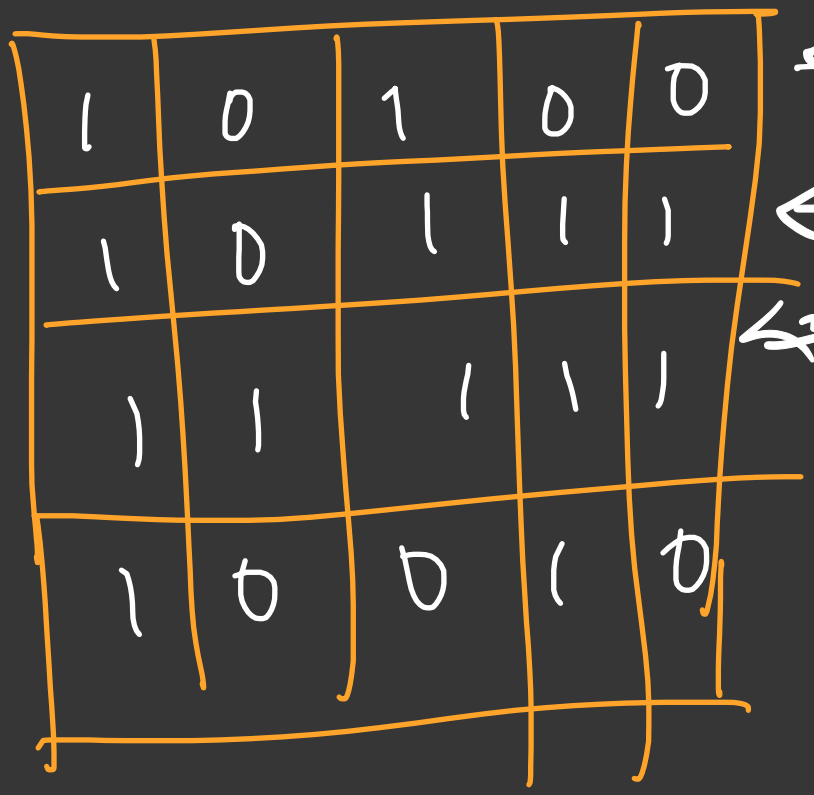
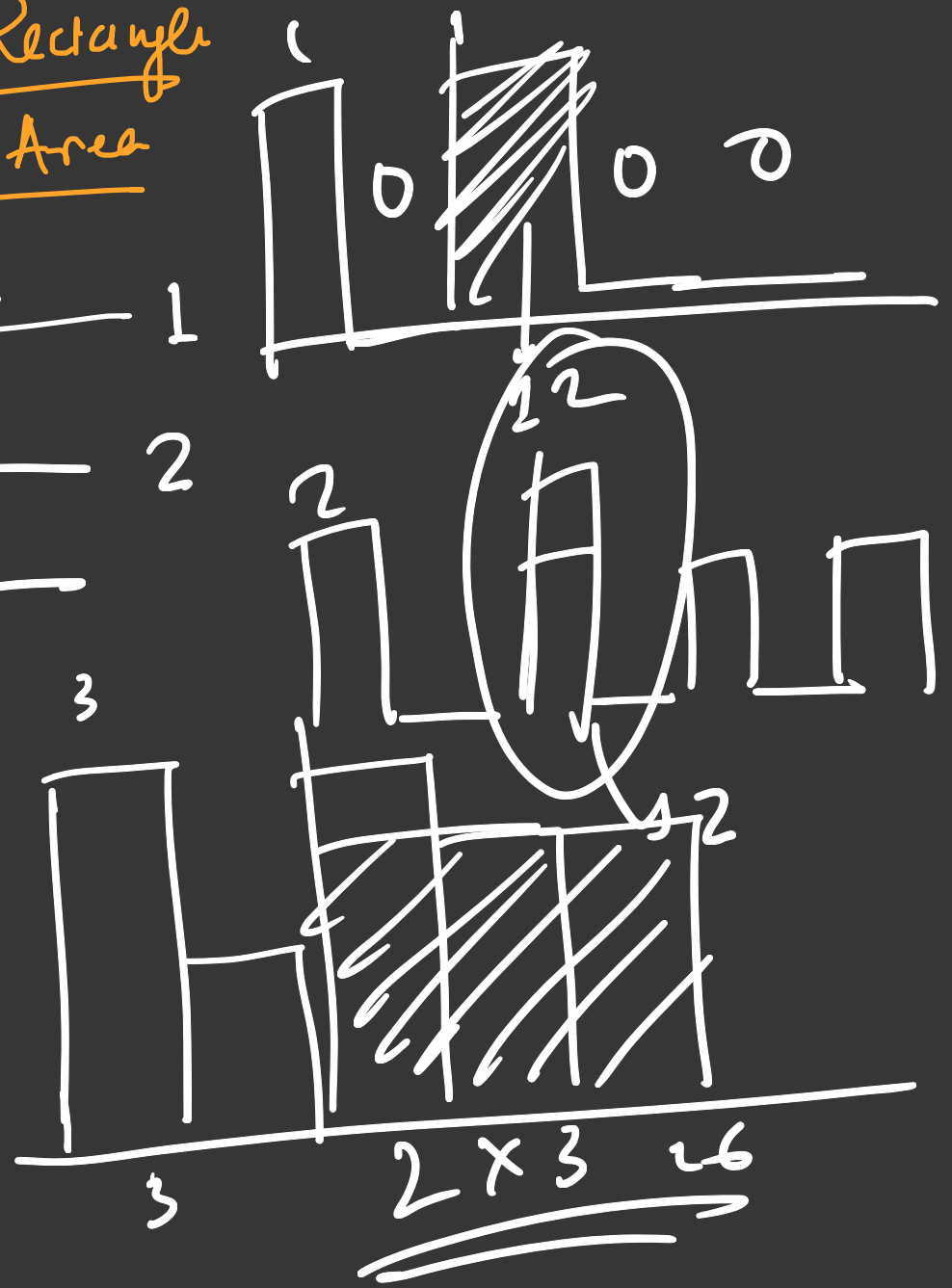
Maximum Rectangle Area





To solve this problem we should know that how to find the maximum area of rectangle in histogram.

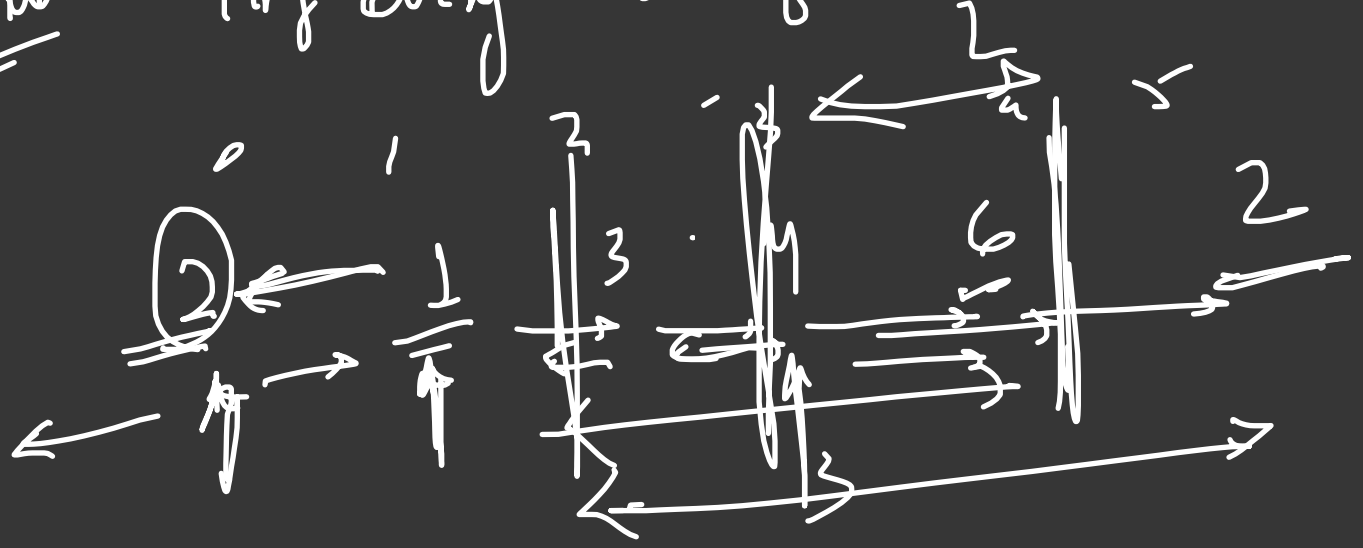
So, in that qn problem

arr[] = { 2, 1, 3, 4, 6, 2 }

3 × 3 = 9

4 × 2 = 8

Naive Try every histogram



2

2 x 2 = 2

for $\frac{O(N)}{O(N)}$

$O(N^2)$

for

2

max(w) 1 x 6 = 6

3 x 3 = 9

4 x 2 = 8

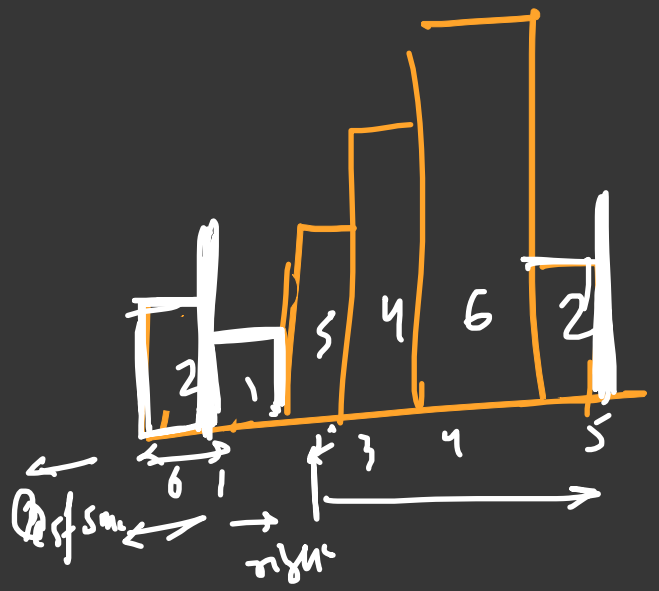
6 x 1 = 6

1 x 2 = 8

2 1 3 4 6 2

leftsmall rightsmall X 2

min = 2



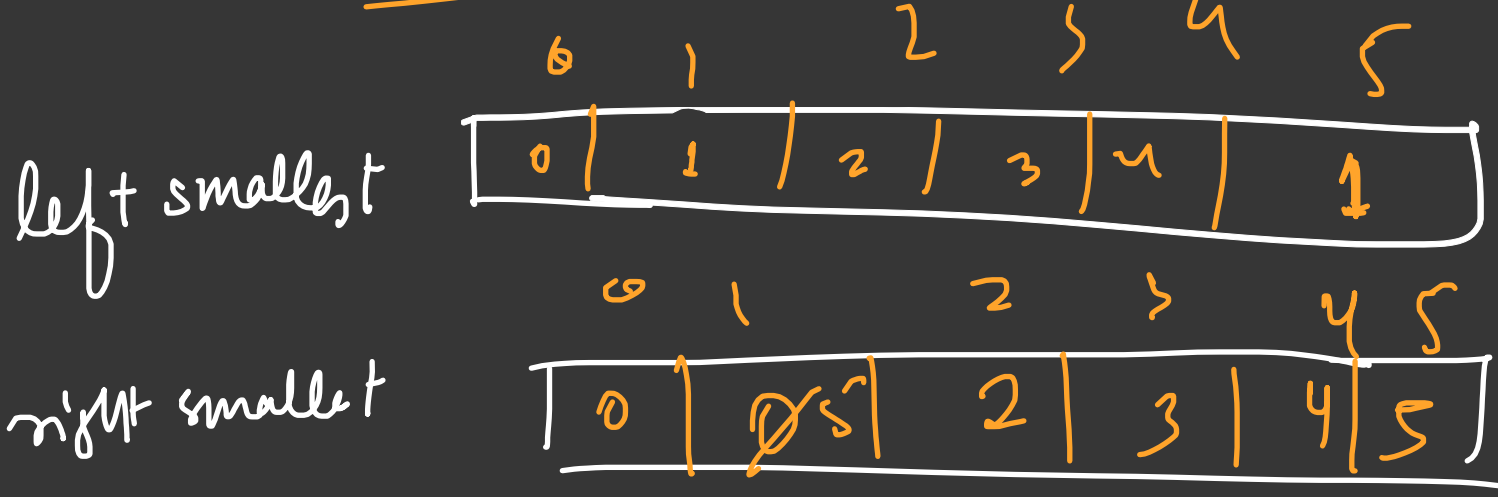
leftsmallest @ right - left =

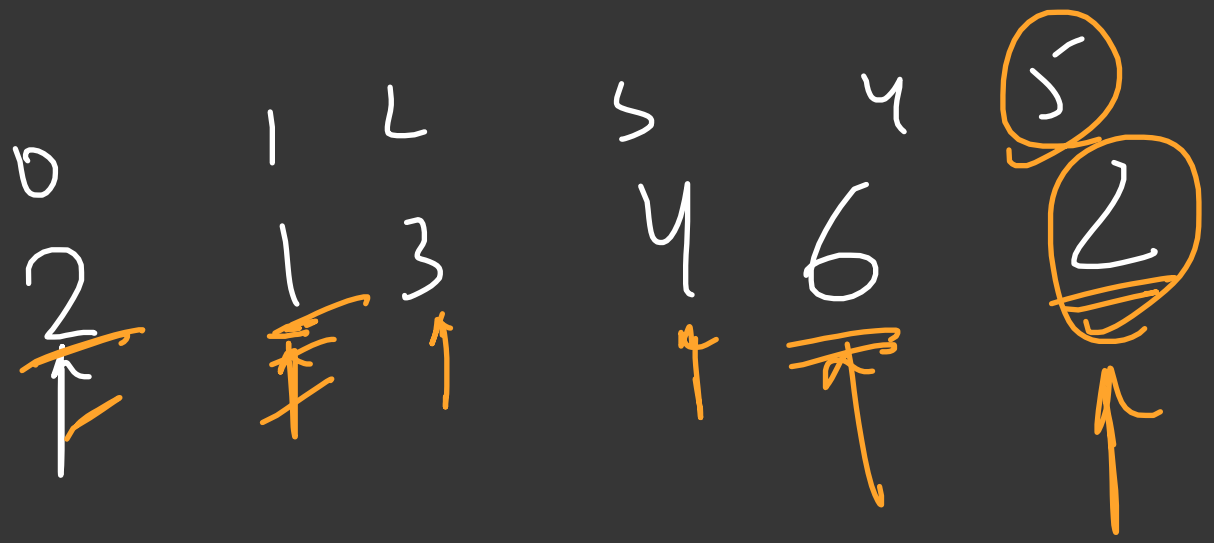
1 - 0 = 1

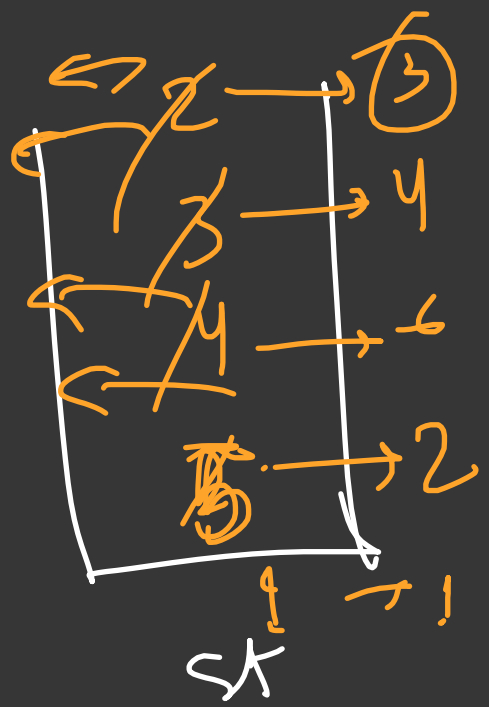
5 - 0 = 2

right left

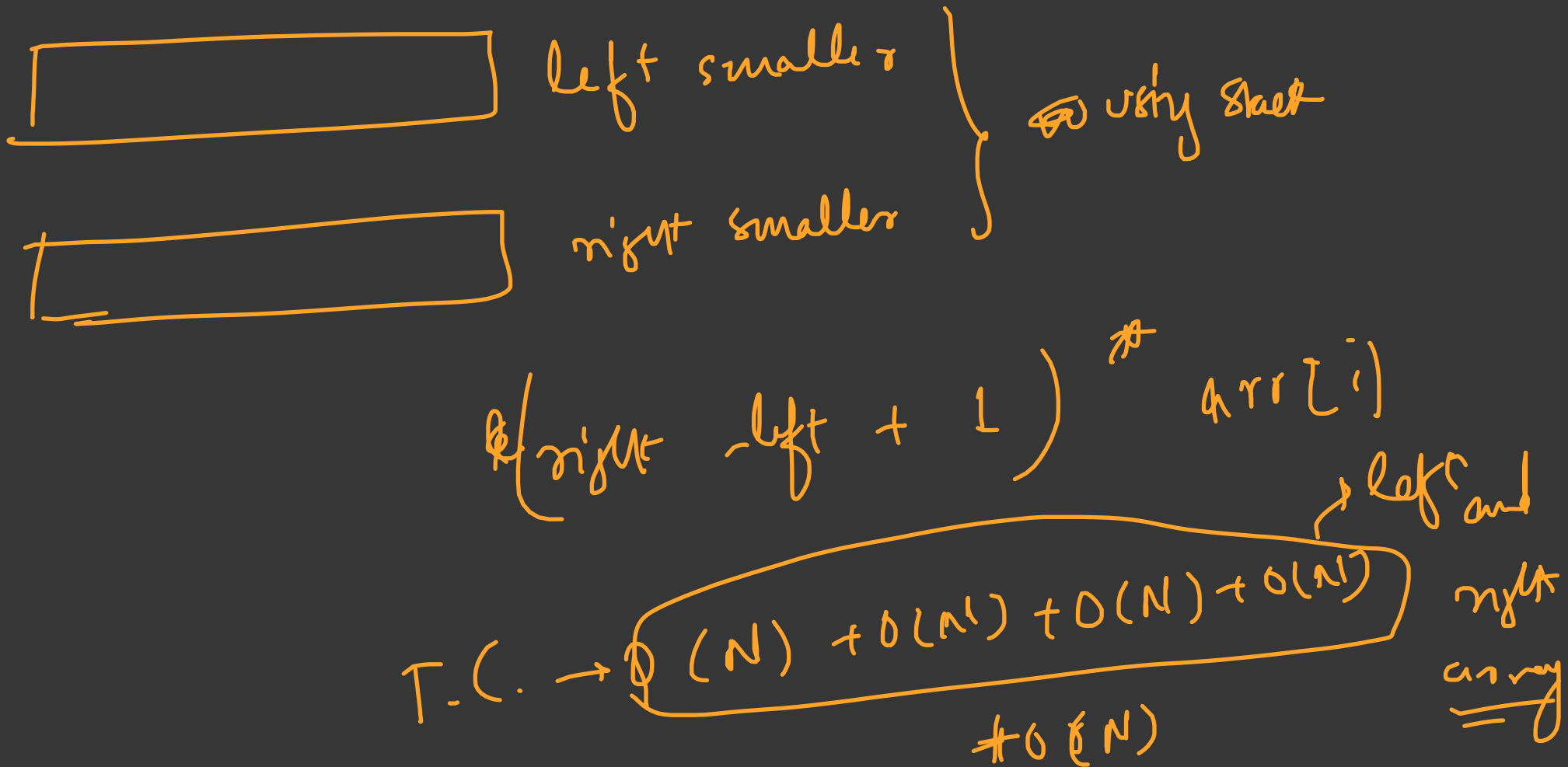
+ ~~left~~ smallest = ~~right~~ smallest * arr[i]



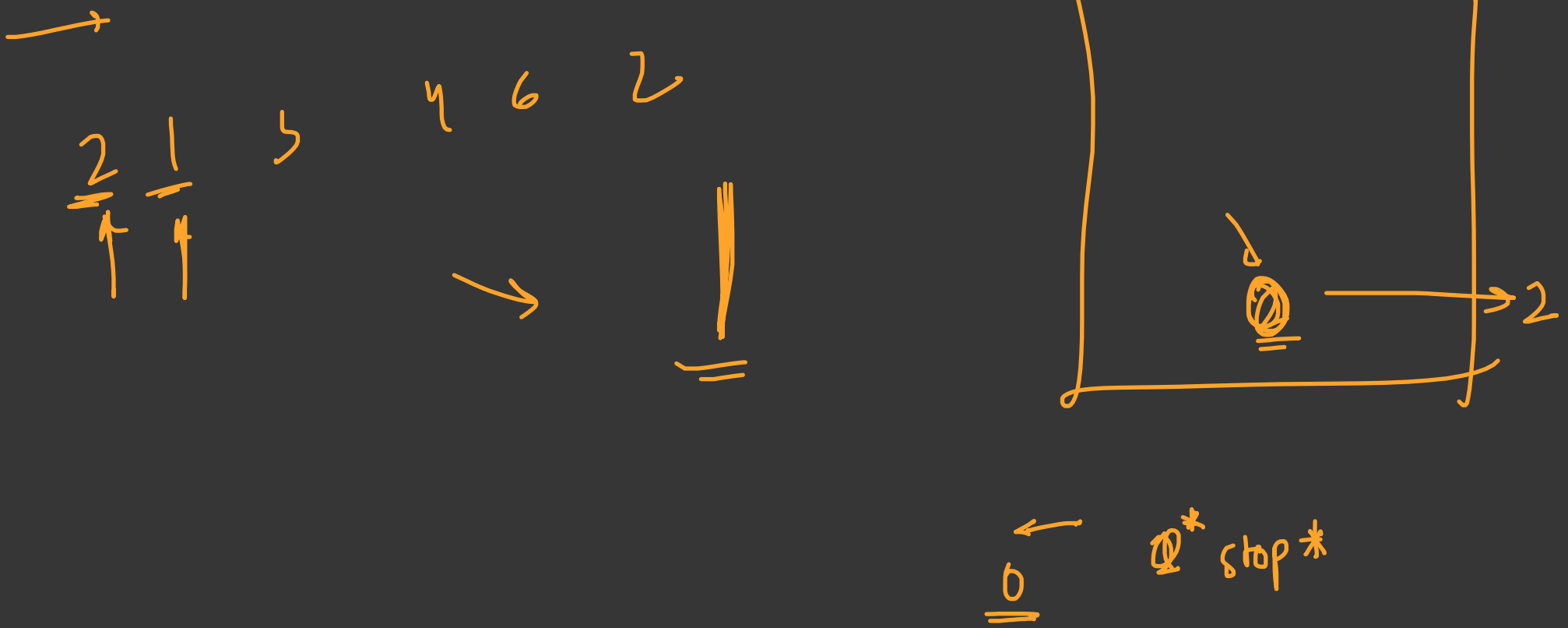


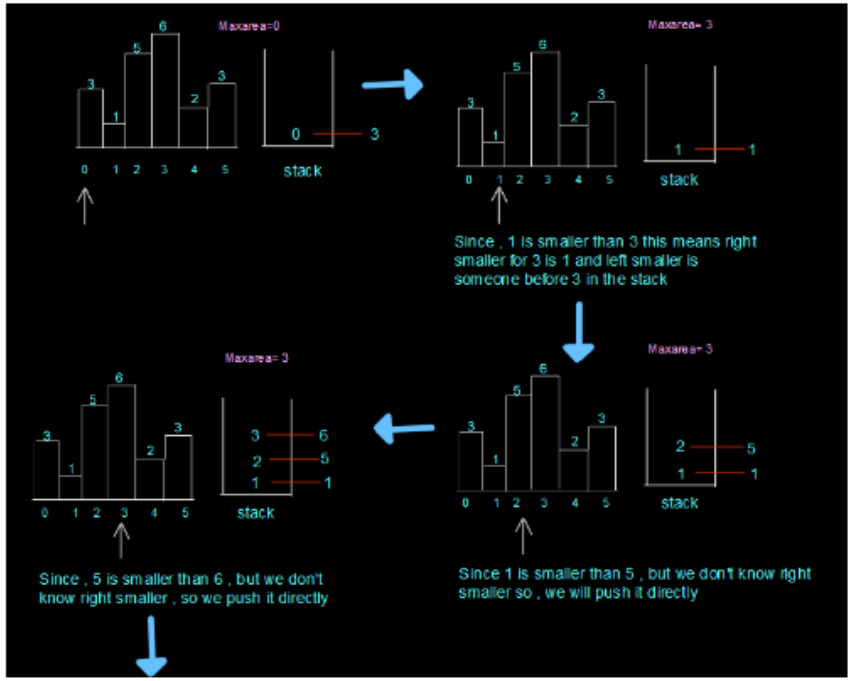


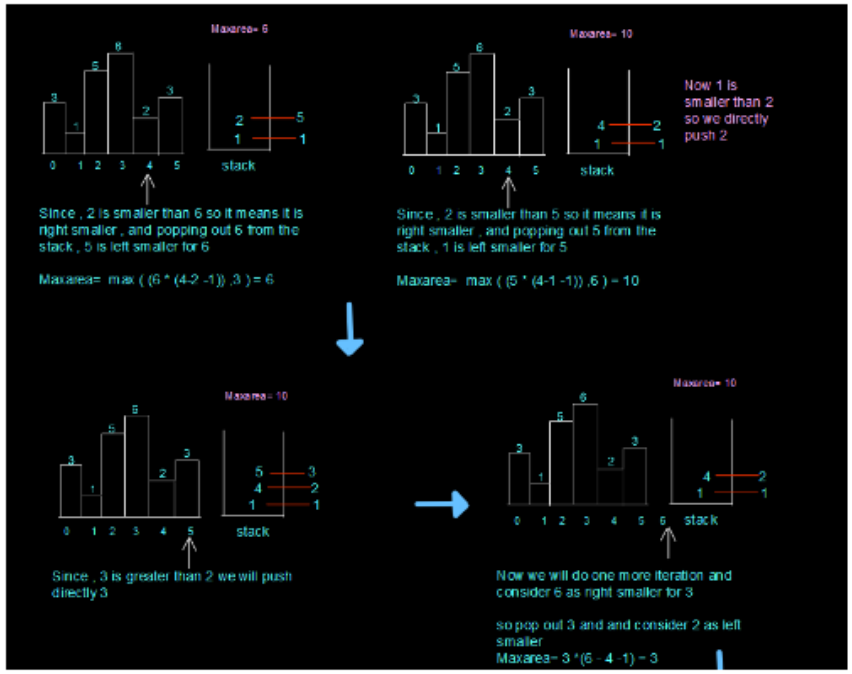
Optimised Approach



We can do it in just 1 pass also?







0 1 2 3 4 5

3 1 5 6 2 3

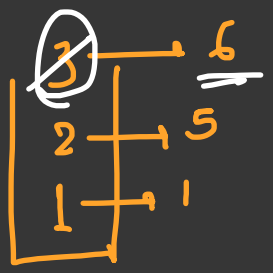
maxA = 6

5 ← 6 → 2

Area = 6 × (4-2-1)

0 ← 3 → right smaller → 1 6 × 1

3 1-0 = 1 × 3 = 3



4 → 2

at which 3 (6) next smaller

2

Pseudo code.

stack<int> st

for (i = 0 ⟶ n)

while(!st.empty && arr[st.top] >= arr[i])

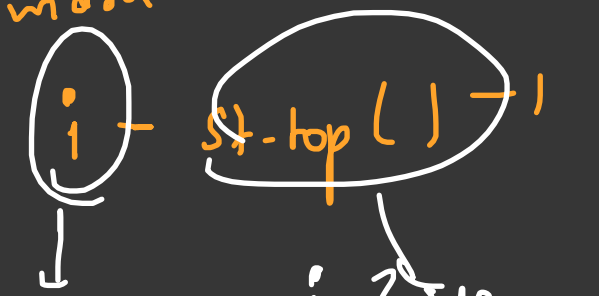
st.pop(),

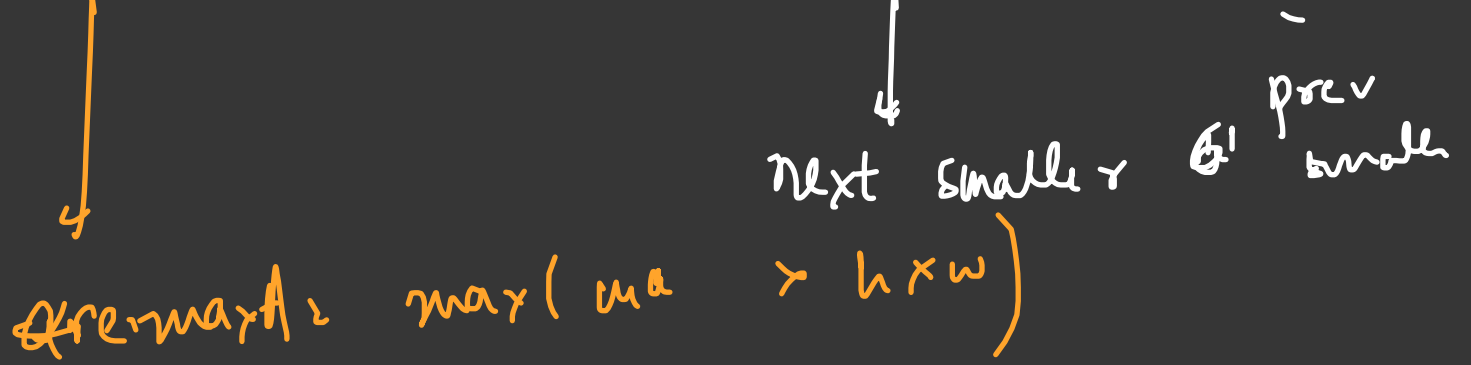
int width;

st.push(i);

if(st.empty) width = i

else width = i - st.top() - 1




Next Smaller
prev smaller
area-max = max(ma
× h × w)

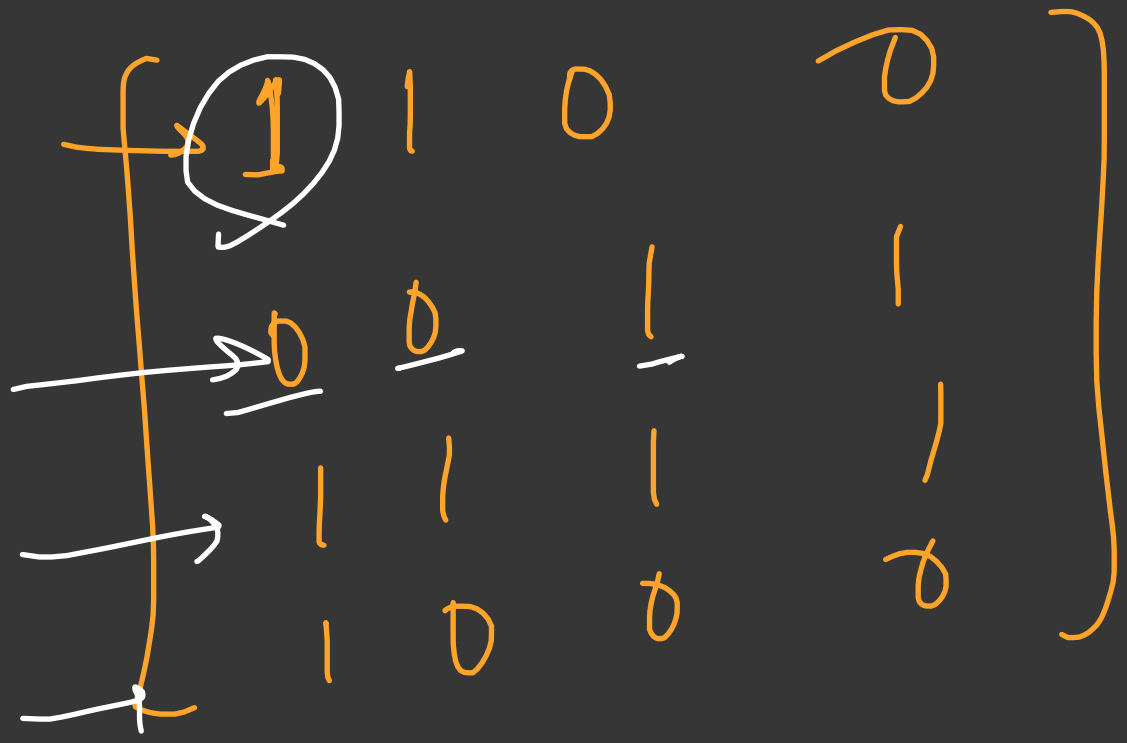
So, for solving the matrix Problem

We visit every row and create a array for the row and calle the function for the row and at every next row we cacld the are length of prev row -

We know that there is 1 & 0 in the matrix

So, if we find a 1 in the matrix we

increase the height.



Pseudocode:

```
for (i = 0 ------> n)
    for (j = 0 ------> m)
        if (mat[i][j] == 20)
            height[j]++;
```