10. Largest Rectangle in Histogram

22 February 2025 23:52

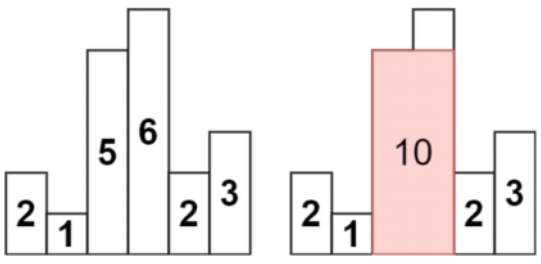
84. Largest Rectangle in Histogram

Solved

Hard | Topics | Companies

Given an array of integers `heights` representing the histogram's bar height where the width of each bar is `1`, return *the area of the largest rectangle in the histogram.*

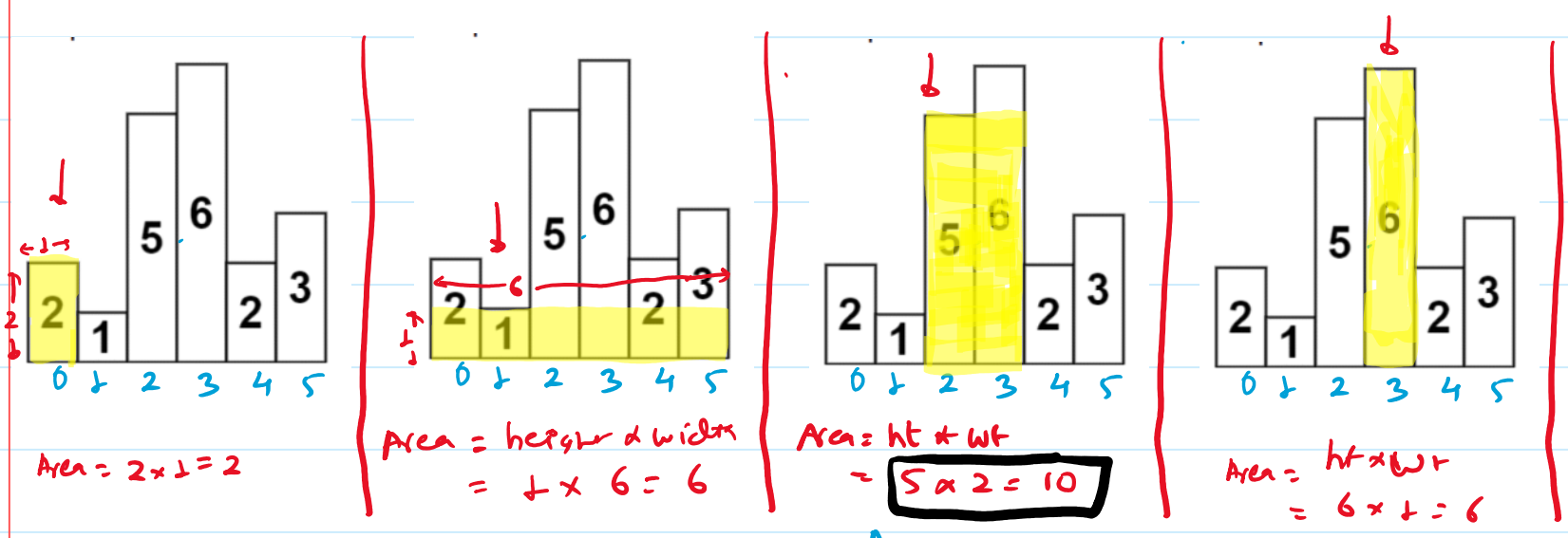
Example 1:

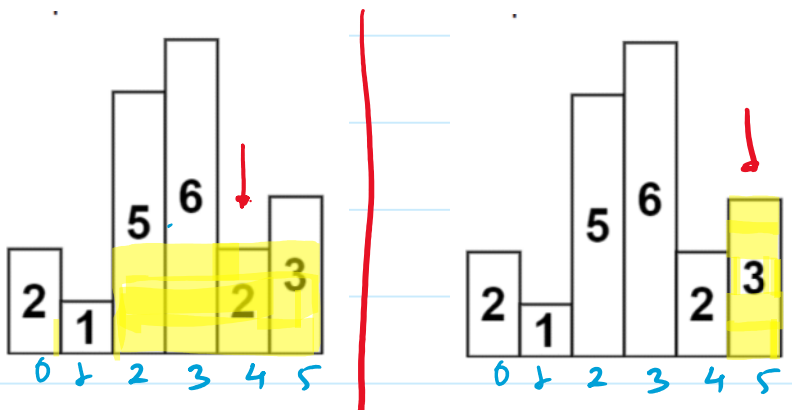



```
Input: heights = [2,1,5,6,2,3]
Output: 10
Explanation: The above is a histogram where width of each bar is 1.
The largest rectangle is shown in the red area, which has an area = 10 units.
```

Method 1: Approach of question

→ Go to each element and to fit what is Maximum Rectangle can form to that index.





Code

— For knowing the (Left) size कहाँ तक element को लेना है।
You observe and find,
previous next smaller element से पहले वाले element.
Similarly
Right side के।
next smaller element से पहले वाले element को लेना है।
— " — prior knowledge
Q NSE (or) PNSE

Formula which Calculate the area.

$$= arr[i] * (NSE - PSE - 1)$$

form (0 → n-1)

Note

if there is No NSE, PSE

(Assign) (-1) (Assign) (n)

Code for this.

```
int fun( arr)
  {
    nSE = find NSE(arr) ;
    PSE = find PSE(arr) ;
      max = 0
    for( i = 0 → n-1)
      {
    max = max( max , arr[i]*(nSE[i] - PSE[i] -1))
      }
  return max ;
  }
```

10. Largest Rectangle in Histogram

23 February 2025 00:24

```java
public int largestRectangleArea(int[] ht) {
    int[] nse=nse(ht);
    int[] pnse=pnse(ht);

    int max=Integer.MIN_VALUE;
    for(int i=0;i<ht.length;i++){
        int a=(nse[i]-pnse[i]-1)*ht[i];

        max=Math.max(max,a);
    }

    return max;
}
```

→ TC O(2N)
→ TC – O(2N)
→ O(N)
TC → O(5N)

SC = O(N) + O(N) } stack
O(N) + O(N) } store arr
= O(4N) → (SC)

```java
public static int[] nse(int[] arr){
    int[] ans=new int[arr.length];
    Stack<Integer> stack=new Stack<>();
    int n=arr.length;
    for(int i=n-1;i>=0;i--){
        while(!stack.isEmpty() && arr[i]<=arr[stack.peek()])
        stack.pop();
        ans[i]=(stack.isEmpty())?n:stack.peek();
        stack.push(i);
    }
    return ans;
}
```

```java
public static int[] pnse(int[] arr){
    int[] ans=new int[arr.length];
    Stack<Integer> stack=new Stack<>();
    int n=arr.length;
    for(int i=0;i<n;i++){
        while(!stack.isEmpty() && arr[i]<=arr[stack.peek()])
        stack.pop();
        ans[i]=stack.isEmpty() ?-1:stack.peek();
        stack.push(i);
    }
    return ans;
}
```

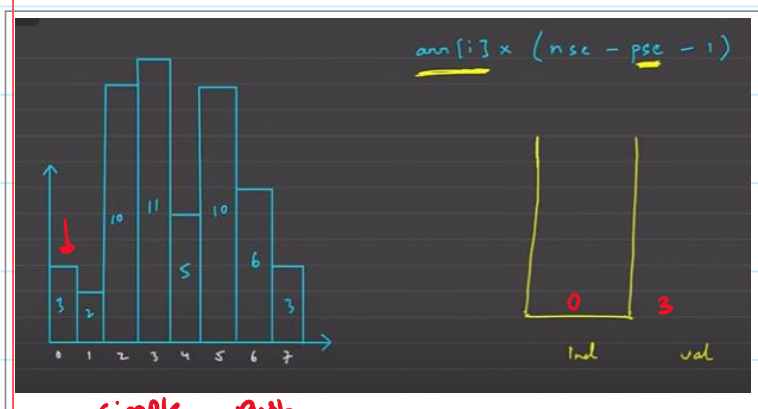
→ O(5N) → Can you reduce O(N) single pass | Method – 2

One thing is sure :

- It should be iterate over At list one time …
- So …let iterate form 0 to N … then … previous smallest element can be find but … difficulty to …find the .. Nse

- For finding the .. NSE … we go back …from the … back side … let's ….. Find how to do these thing …

  1. I traverse then .. Return back to element when get … NSE …. And … previous smallest element … can already find the same concept



Simple – push

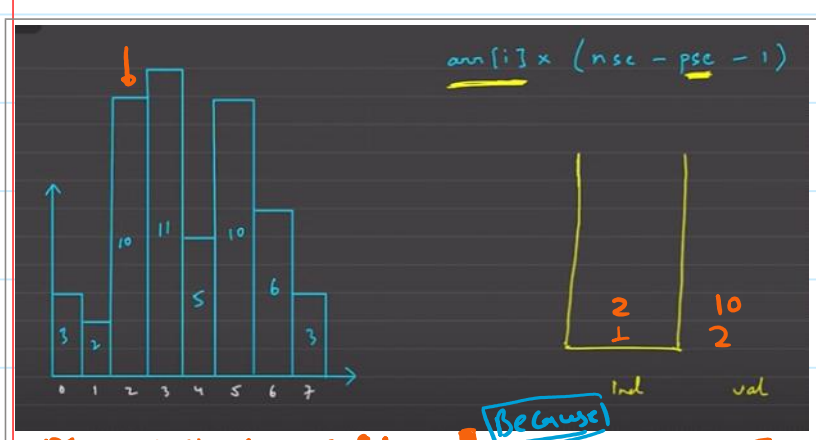
this is Curr element
3 ← pop()

2 is the NSE of 3
x (NO) is the PSE of 3
↳ Consider (-1)

arr[stack.pop()] * (nse - pse - 1)
3 * (1 - -1 - 1) = 3 ⇒ 3 Area
↑ Current idx

max = 3

At last push on stack
Current (idx)
i.e

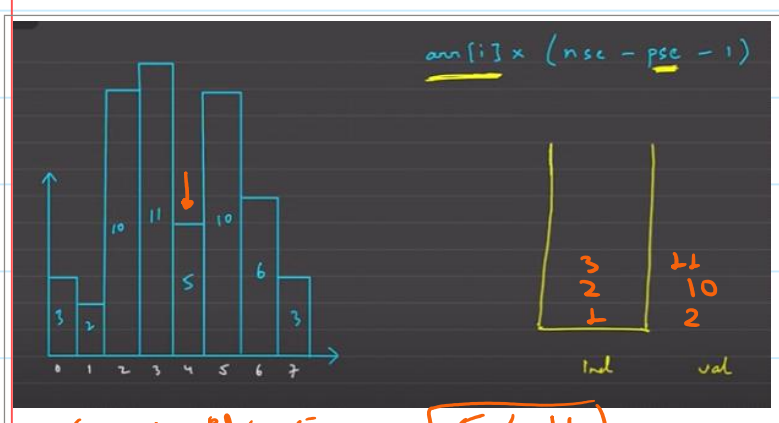


Direct push Current idx = 2 Because arr[curr] > arr[peek]

→ push idx = 3

10. Largest Rectangle in Histogram

23 February 2025 00:54



Current idx = 5, $\boxed{5 < 11}$ → POP the Stack

i.e
5 is NSE of 11 → $\boxed{POP()}$
10 is PSE of 11 →
→ i.e stack के निकले वाले

Area = 11 * (4 - 2 - 1)
11 * 1 = 11

i.e max = ~~3~~ 11
→ And pop out → (11) el idx.

Curr = idx = (4) element = 5

$\boxed{5 < 10}$
→ stack.el
curr.el.

5 is NSE of 10; →
2 is PSE of 10; → (10) pop out

then area is → 10 * (4 - 1 - 1)
10 * (2) = 20

Max = ~~11~~ 20
→ Now the current idx push (4)
→ el = 5

→ push curr idx

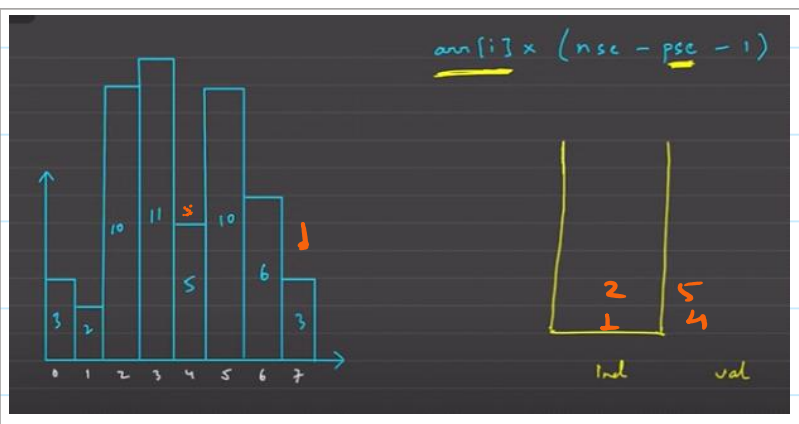
→ curr el (6 < 10)
pop exit

curelement : 10 (id = 5)
nse = 6 (idx = 6)
pse = 5 (pidx = 4)

Area = 10 * (6 - 4 - 1) → 10 * 1 = 10
Max = 20 > 10 (No update)
→ push 6 idx on stack

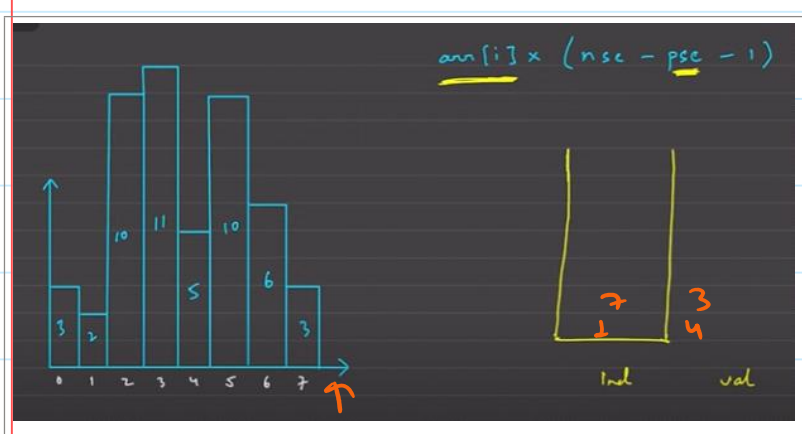
Curr = 3 then (3 < 6) pop
Curr = 6
nse idx = 7
pse idx = 4
Area = 6 * (7 - 4 - 1)
6 * 2 = (12)

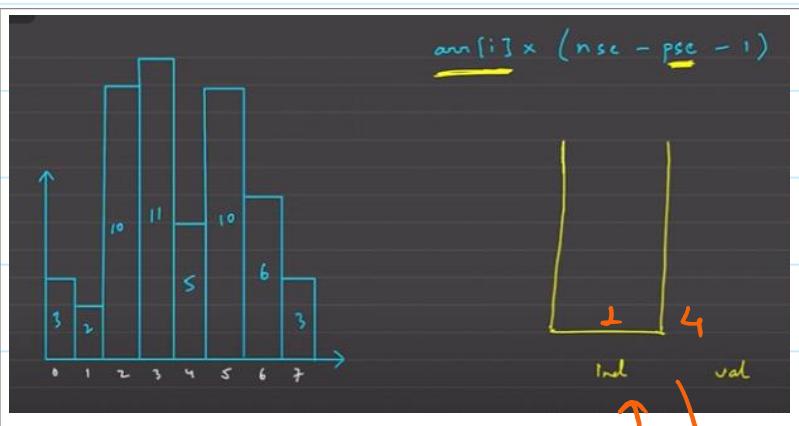
→ Not update the Max value.
→ (3 < 5)



3 < 5

Curr = 5, nse idx = 7, pse idx = 1.
Area = 5 * (7 - 1 - 1) = (25)
→ Now push (7) idx





Consider Hypothesis NSE arr = n (here is 8)
→ if (stack contains element
→ then pop करके → peek() → PSE.

For element (3) idx = 7.
3 * (8 - 1 - 1) = 3 * 6 = (18) Not update
POP()
Only last

nse idx = 8, curr = 2, prev = (-1) stack is Empty
i.e Area = 2 * (8 - -1 - 1) = 2 * 8 = (16) || Not updated

10. Largest Rectangle in Histogram

23 February 2025 01:24

Code

Stack<Integer> stack=new Stack<>();
int ans=0, n=arr.length;
for(int i=0;i<n;i++){
while(!stack.isEmpty() && arr[i]<arr[stack.peek()]){
int curr_val=arr[stack.pop()];
int nse=i;
int pnse=(stack.isEmpty())?-1:stack.peek();
ans=Math.max(curr_val*(nse-pnse-1),ans);
}
stack.push(i);
}
while(!stack.isEmpty()){
int curr_val=arr[stack.pop()];
int nse=n;
int pnse=(stack.isEmpty())?-1:stack.peek();
ans=Math.max(curr_val*(nse-pnse-1),ans);
}
return ans;
}
		TC- 0(N) +0(N)