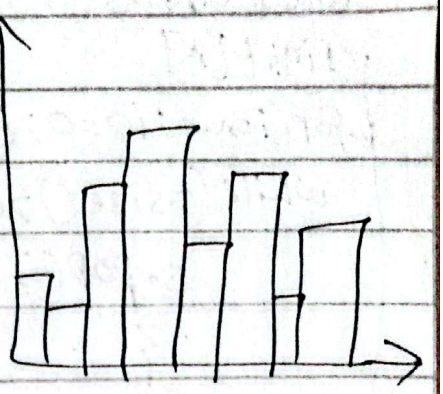
Largest Rectangle in Histogram

height = [2,1,5,6,2,3]



→ Return largest rectangle area

Area = ht * width

↓

Maximum

(1) Brute Force :- $O(n^2)$

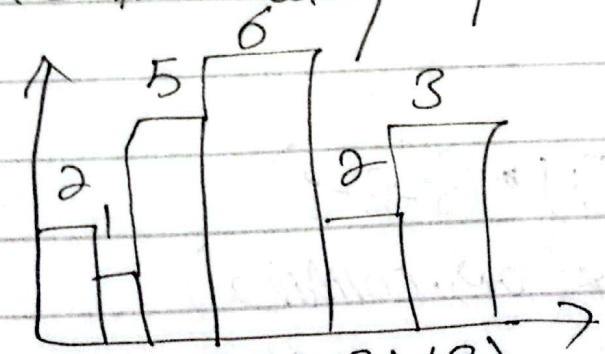
→ Find all the rectangles areas and return largest rectangle area.

Outer loop (Start)

Inner loop (End)

(2) Optimal Approach $O(n)$

Make the rectangle for each bar.



Area → (2)(6)(10)(6)(8)(3)

width

currArea = height[i] * (i - l - 1)

left smaller nearest ← → right smaller nearest

Find left smaller nearest ← (Reverse Traversal) → find right smaller nearest

Right Smaller Nearest

1	-1	4	4	-1	-1
0	1	2	3	4	5

```
stack<int>s
right[n]
for(i=n-1;i>=0;i--){
while(s.size()>0 && ht[s.top()]>=ht[i]){
    s.pop()
    }
right[i] = s.empty()?-1;
    s.top();
    s.push(i)
```

Left Smaller Nearest

-1	0	1	2	4

```
left[n]
for(i=0 to n)
    width = i-l-1
```

right[i] left[i]

```
ans=0
for(i=0 to n)
currarea = height[i] * (r-l-1)
ans = max(ans, currArea)
    }
return ans
```

T.C = O(n)

S.C = O(n)