Next Greater Element

4 5 2 25

4 → 5

5 → 25

2 → 25

25 → -1

Brute force Approach.

T.C → $O(n^2)$

4 5 2 25

5	25	25	-1

Worst Case

1000 999 998 997 ... 1 2000

i = 2

j = 2 - - -

j

NGE

2000	2000	2000	2000	2000	...

1001th times 1000th 999th times

arr = n $n-1$ $n-2$ — — — — — 1 $n+1$

$i=1$

$j=1$

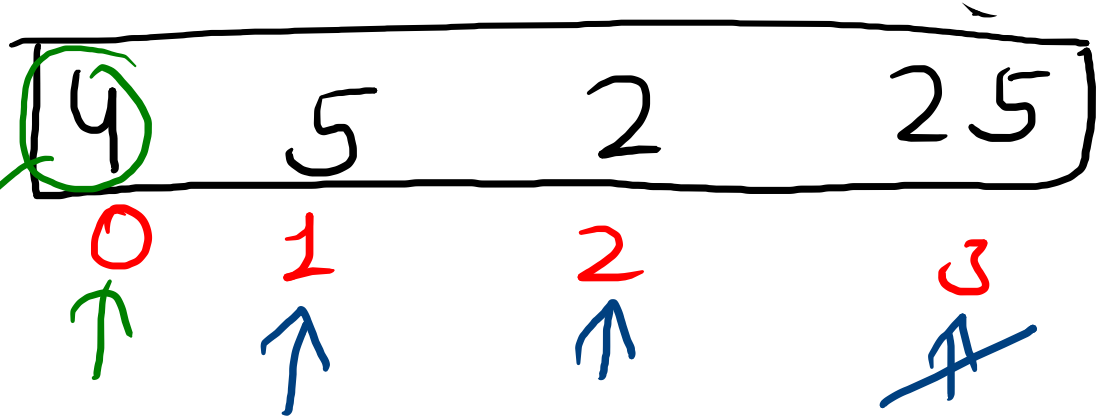
NGF

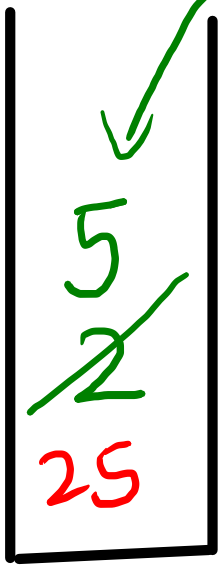
n^{th}	$(n-1)^{th}$	$(n-2)^{th}$ — — — 1

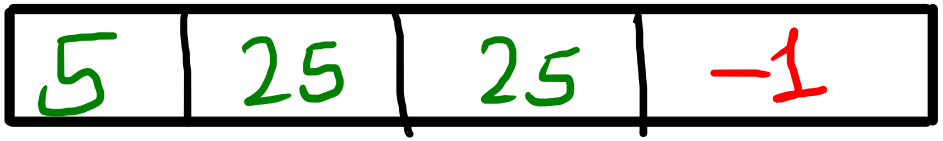
$$n + (n-1) + (n-2) + \cdots 1$$

$$= \frac{n(n+1)}{2} = O(n^2).$$

Stack.



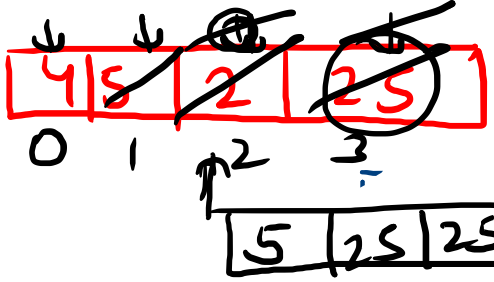


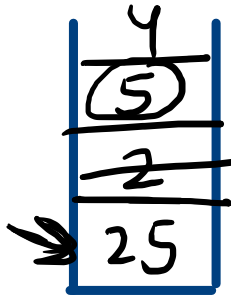


```
NGE()
{
  Stack s;
  nextG[];
  for(i=arr.length-1; i>=0; i--)
  {
      while(!s.empty && s.peek() <= arr[i])
      {
          s.pop();
      }
  }
}
```

O(n)

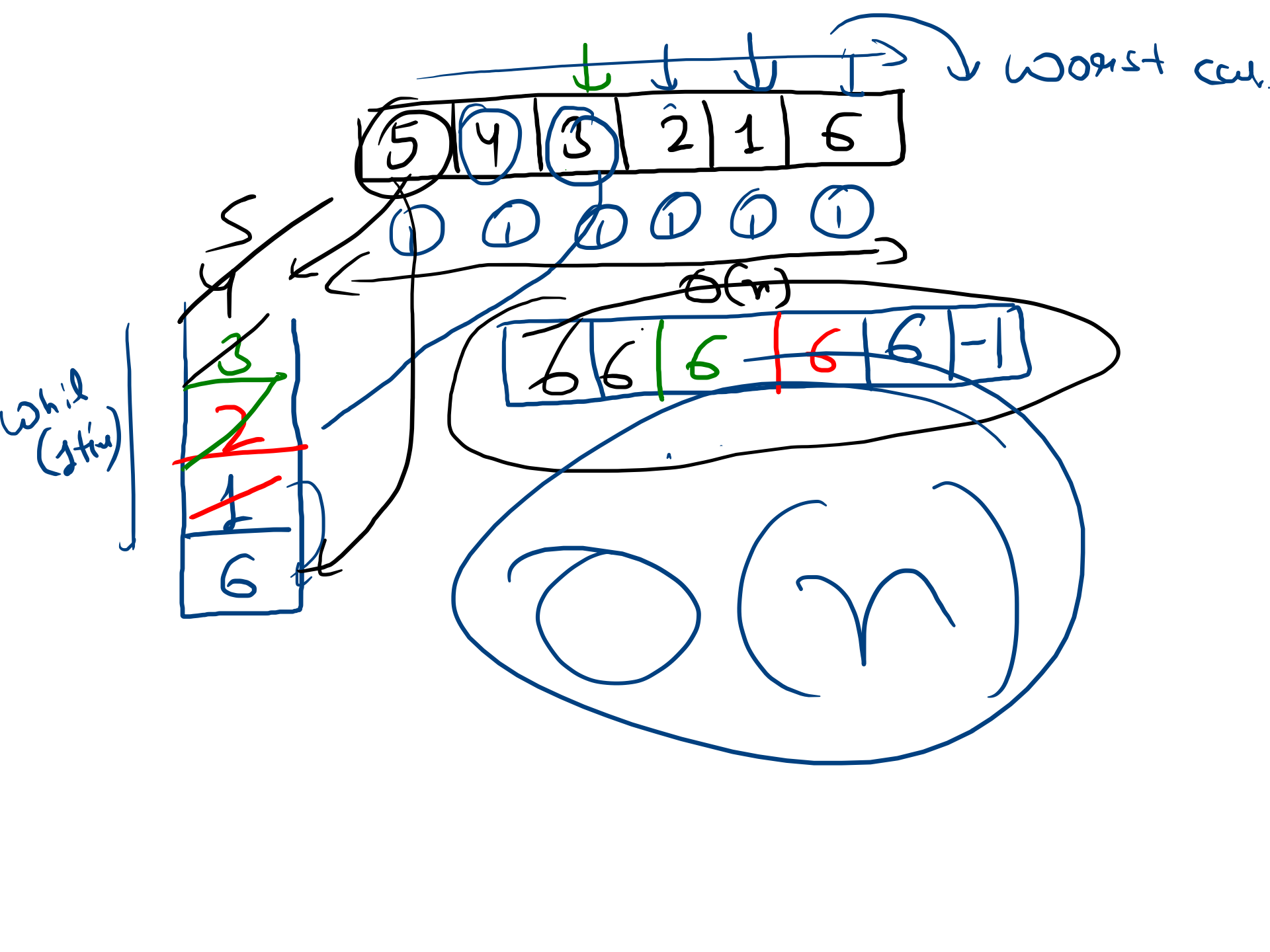
```
NGE(){
  Stack s;
  nge[];
  for(int i = arr.length-1; i>=0; i--)
  {
    while(!s.empty() && s.peek()<=arr[i])
      {  s.pop();
      }

    if(s.empty) { nge[i] = -1; }
      else {nge[i] = stack.peek();}

    s.push(arr[i])
  }
```

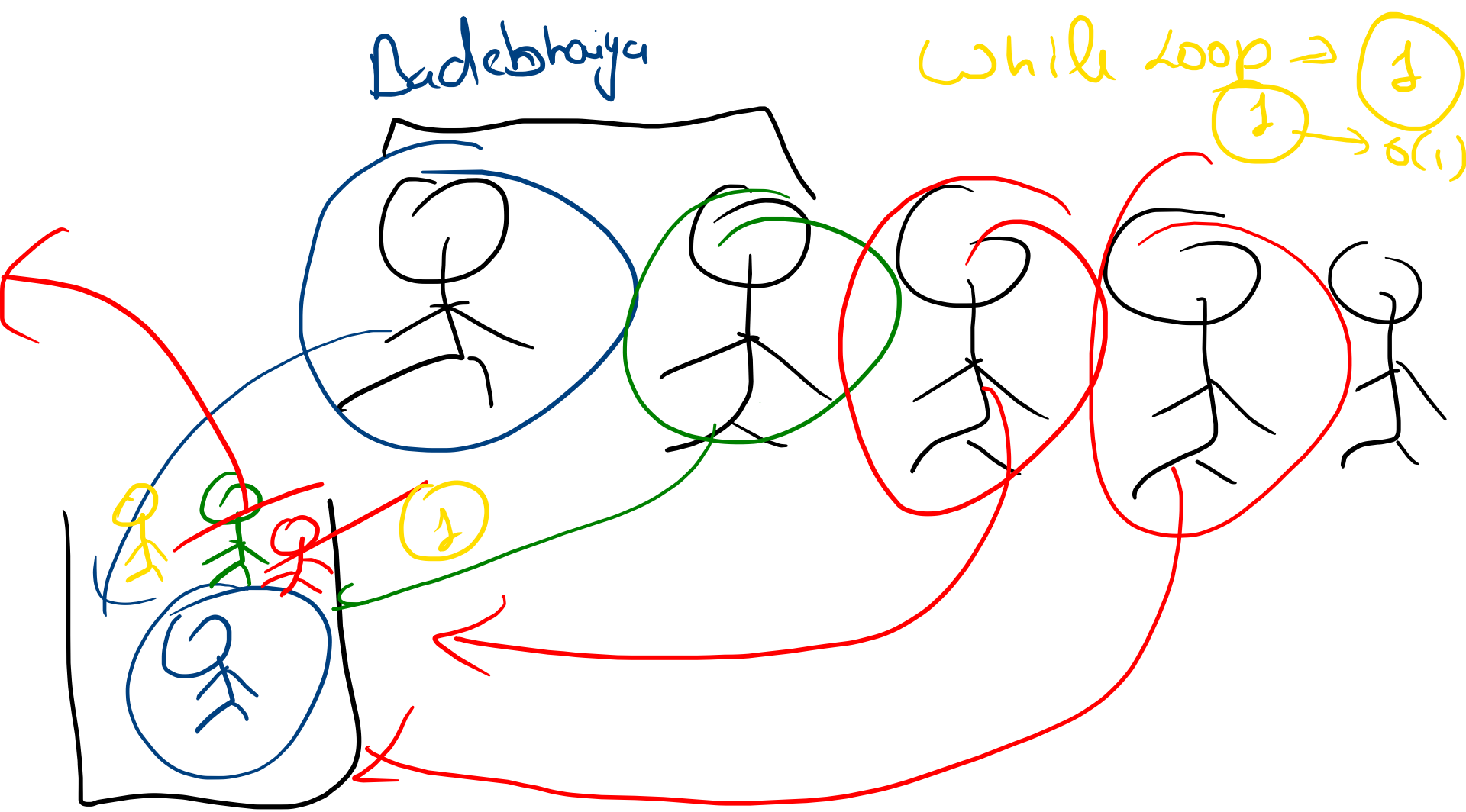




s.{ }

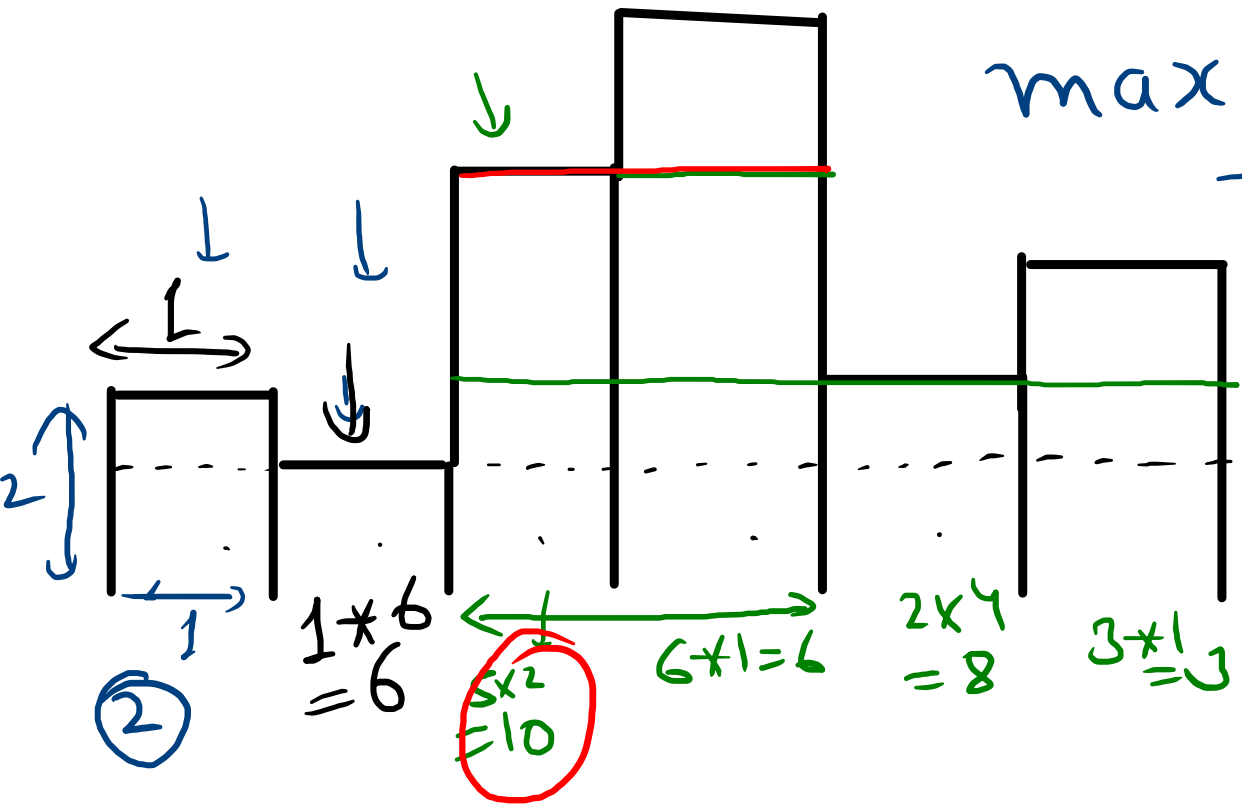
O(n)


Worst case
5 4 3 2 1 6
1 1 1 1 1 1
O(n)
6 6 6 6 6 -1
5
4
3
2
1
6
While (stk)


Dadebhaiya
while loop
1
1
O(1)
1

Maximum rectangle area in histogram.

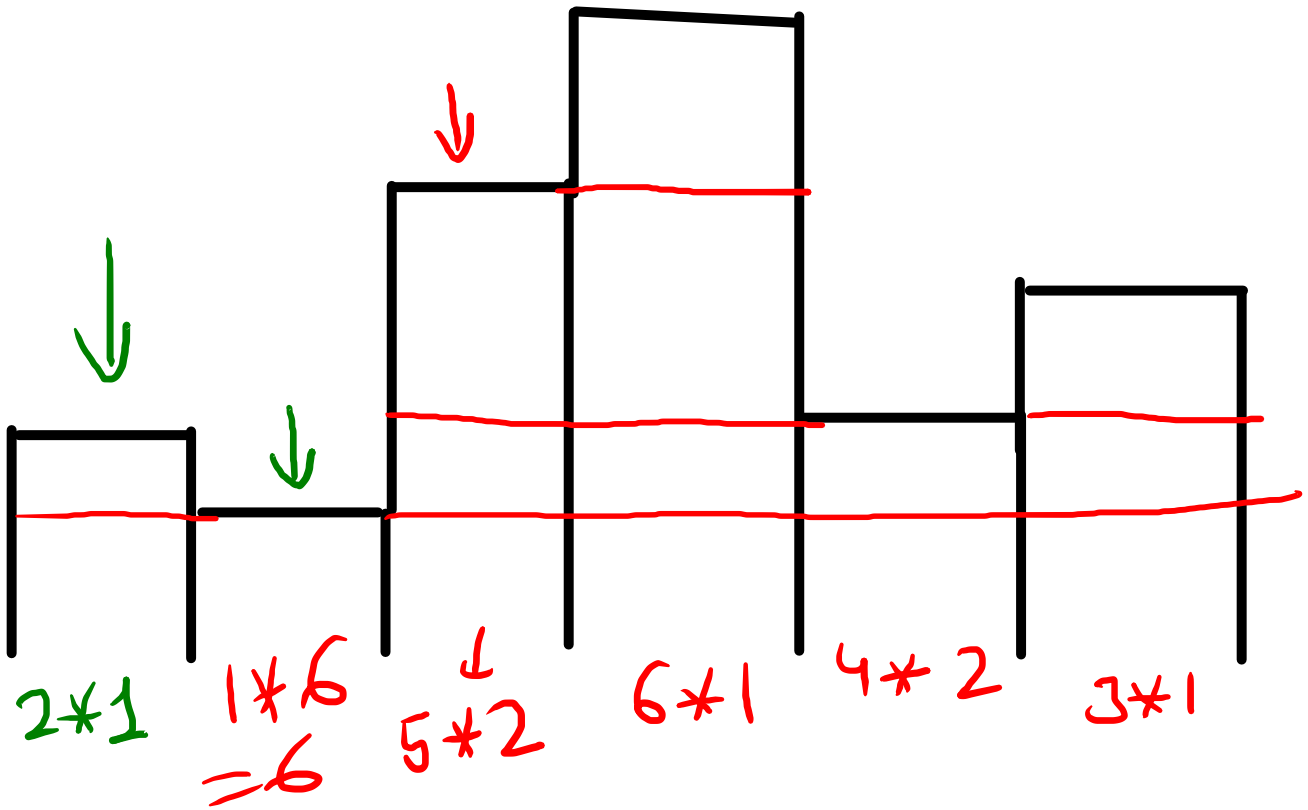
2 1 5 6 2 3



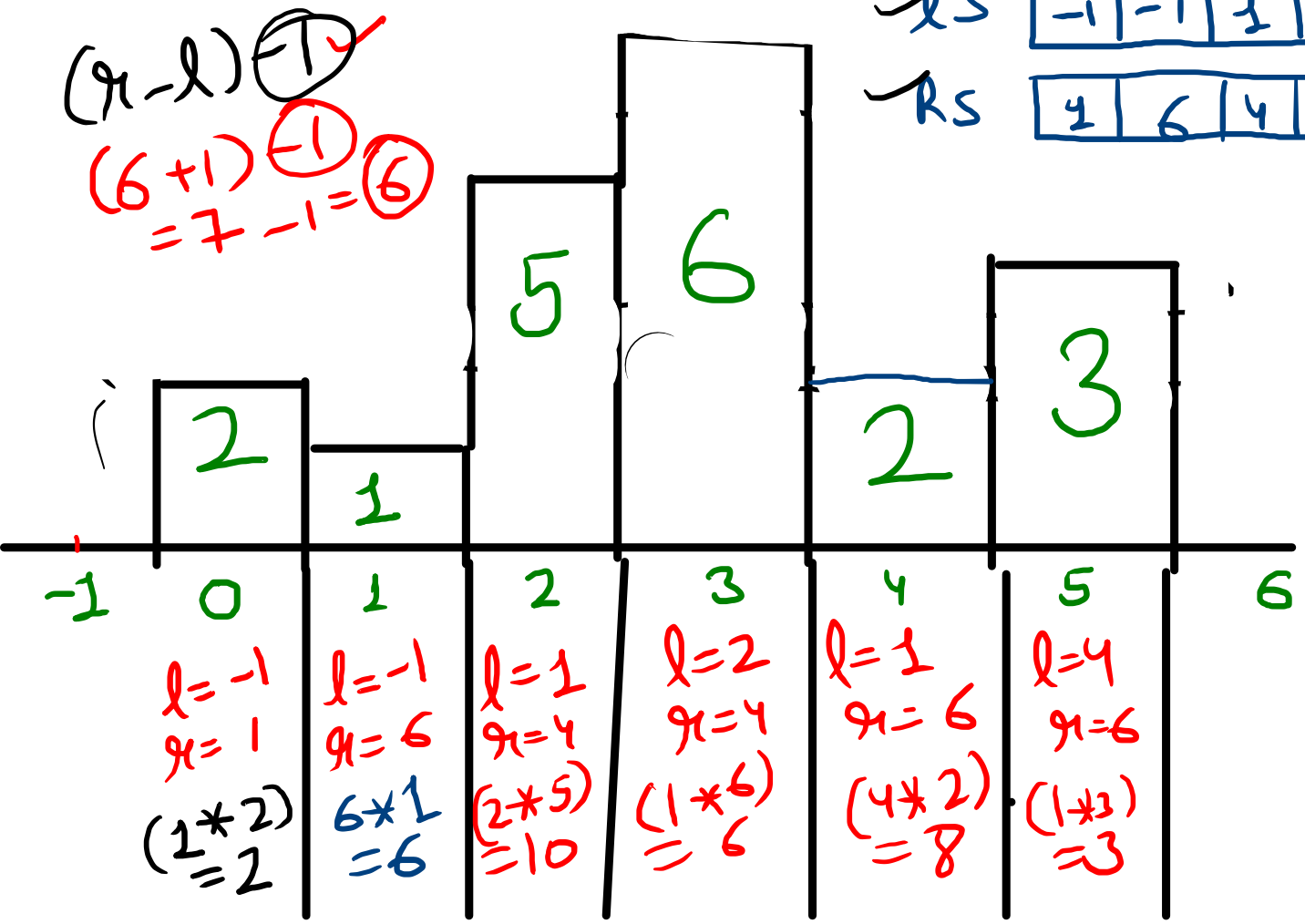
max(2, 6, 10, 6, 8, 3)
= 10

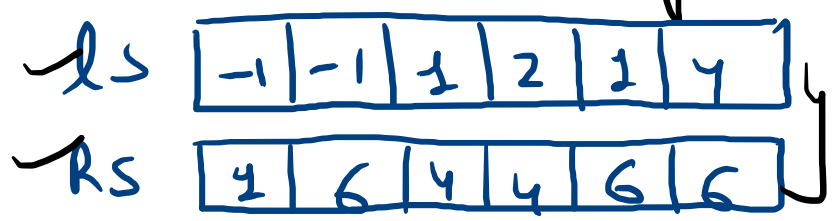
Maximum rectangle area in histogram.

2 1 5 6 2 3



Maximum rectangle area in histogram.





indexes-

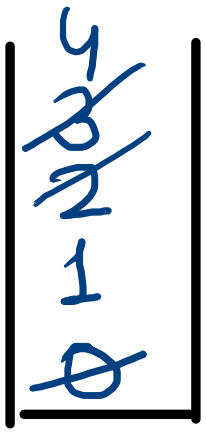
Left Next Smallest

Right Next Smallest

Smallest to left.

Smallest → Chota bhai

2	1	5	6	2	3
0	1	2	3	4	5

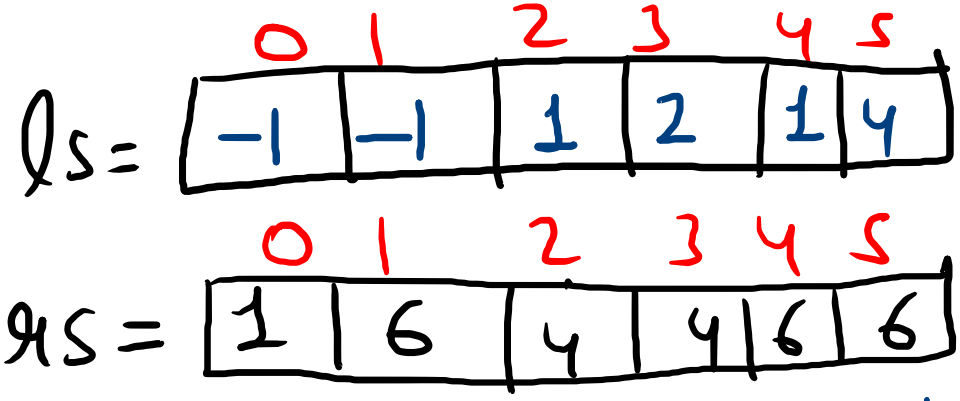


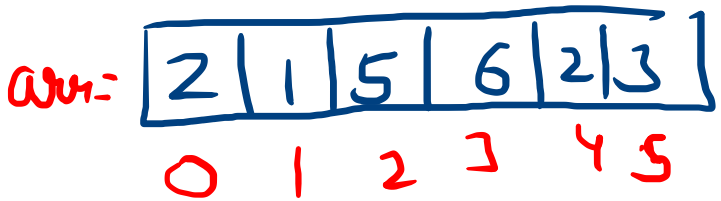
→ O(n)

ls = | -1 | -1 | 1 | 2 | 1 | 4 |

rs = | 1 | 6 | 4 | 4 | 6 | 6 |

→ O(n)





max (1*2, 6*1, 2*5, 1*6, 4*2, 1*3)

= 10

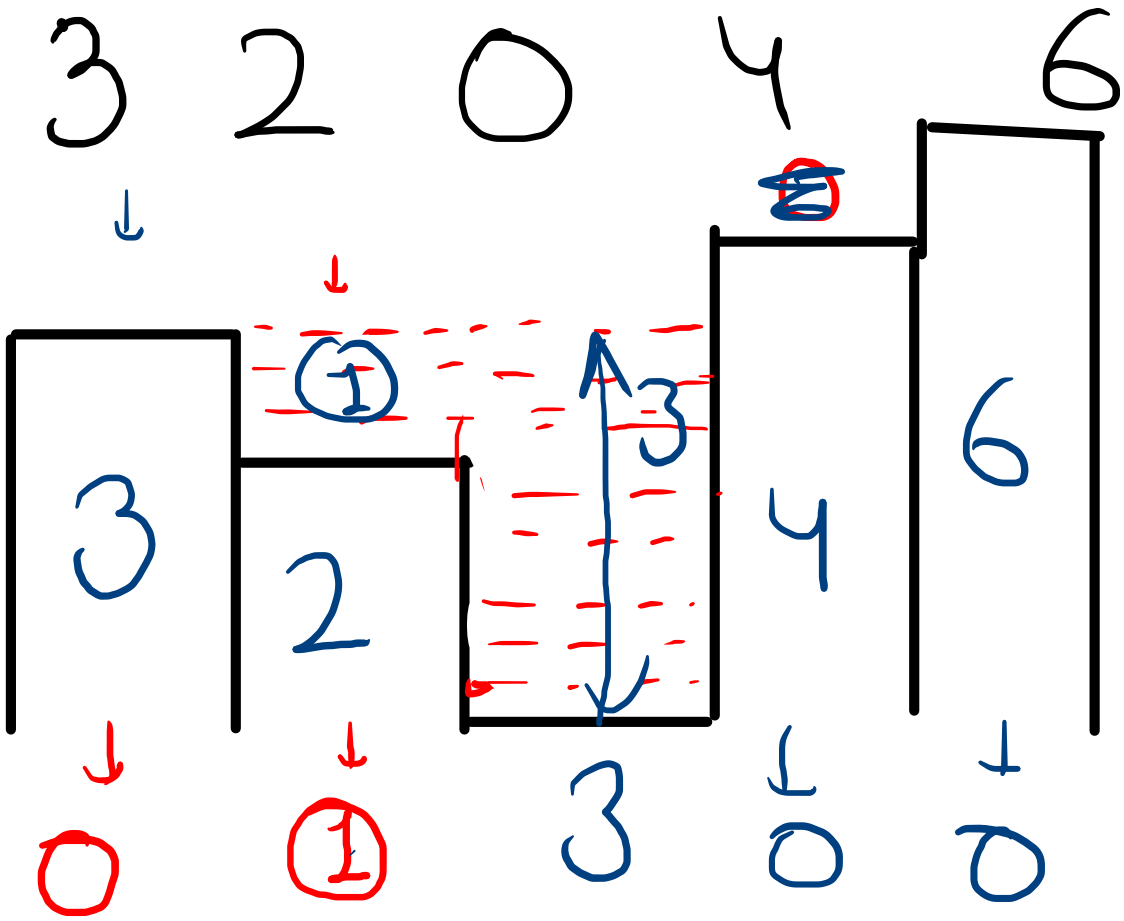
O(n)

((rs[i] - ls[i] - 1) * arr[i]) — max

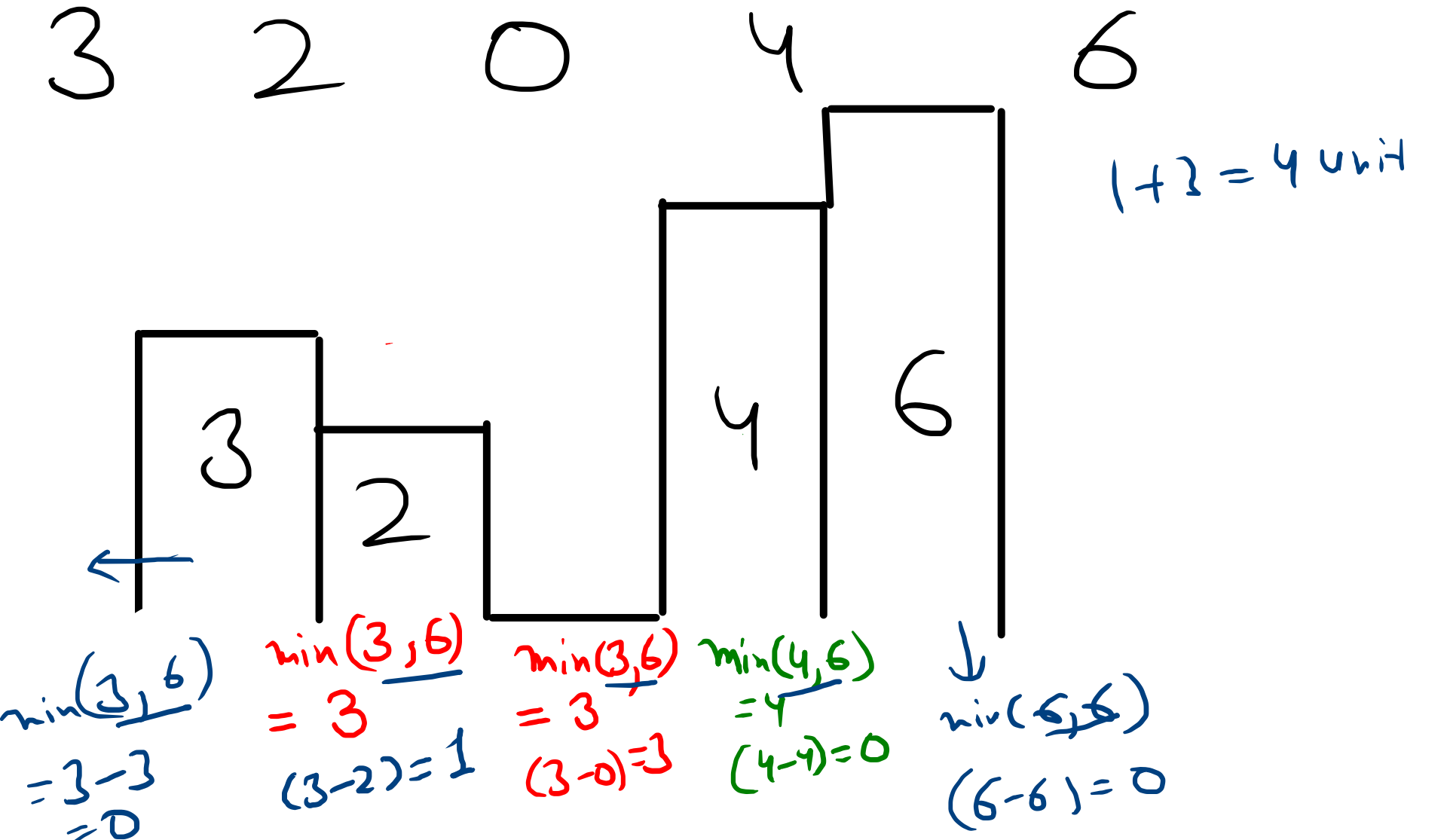
$$O(n) + O(1) + O(1)$$

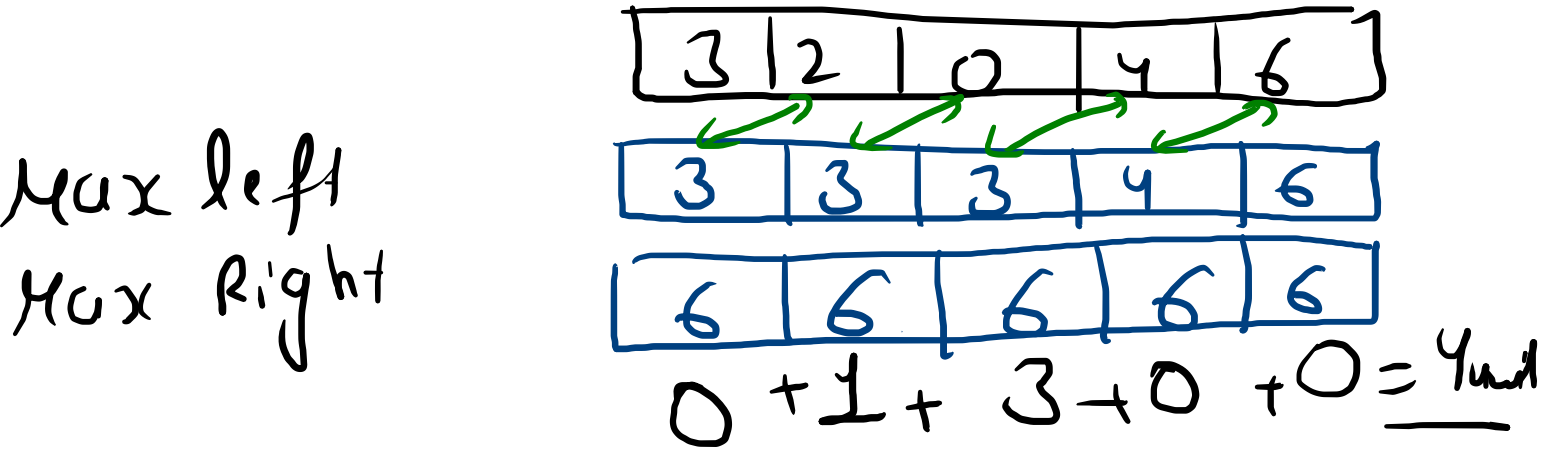
$$= O(n)$$

Rain water problem.

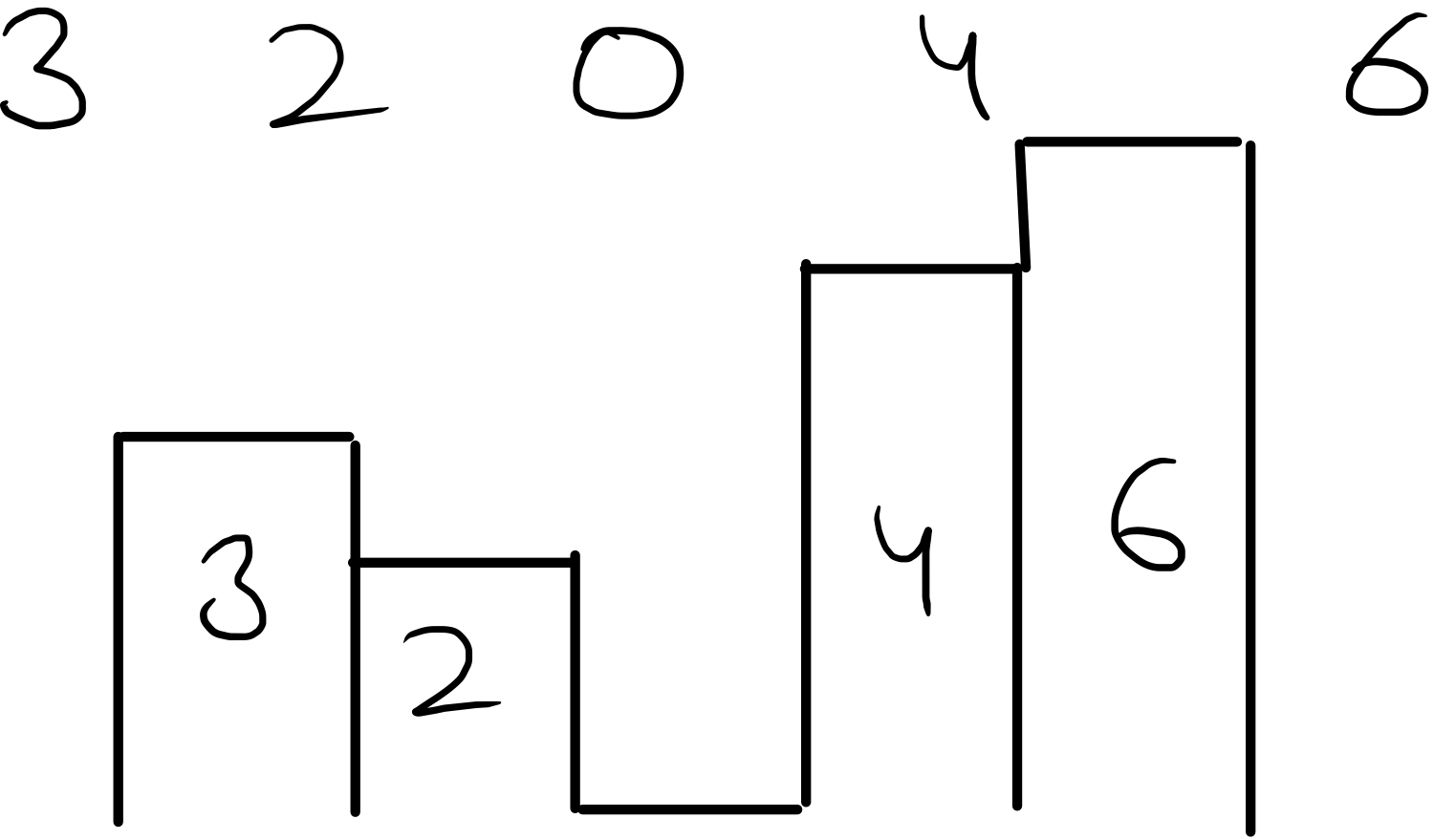


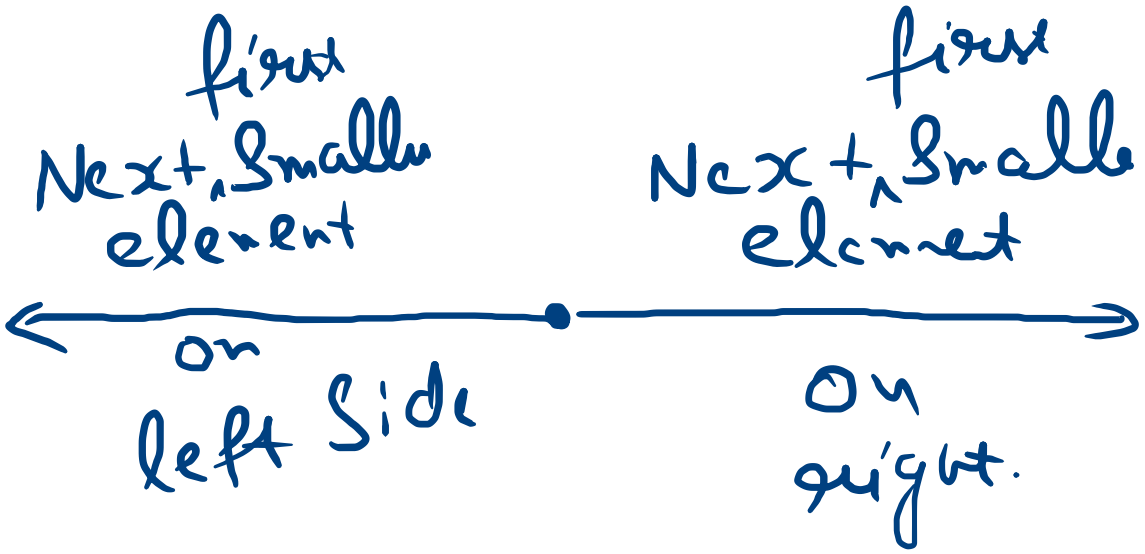
$= 3 + 1 = 4$ unit.


3 2 0 4 6
1+3 = 4 unit
3
2
4
6
min(3,6) = 3 (3-3) = 0
min(3,6) = 3 (3-2) = 1
min(3,6) = 3 (3-0) = 3
min(4,6) = 4 (4-4) = 0
min(6,6) (6-6) = 0



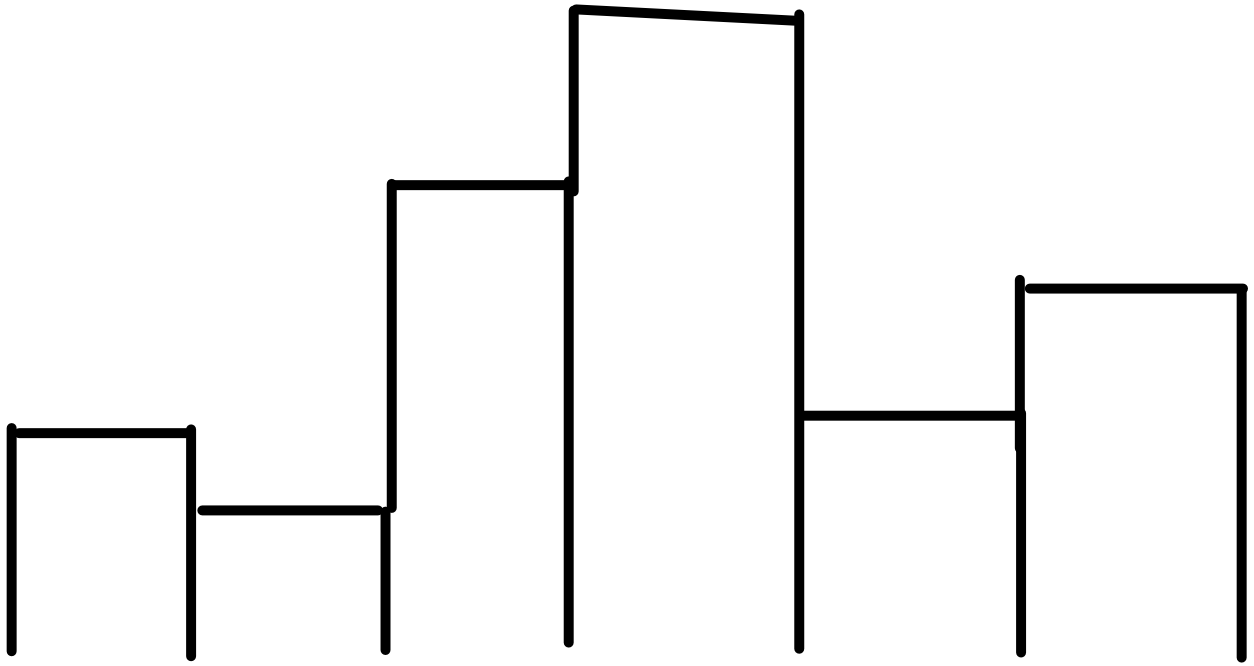

```
for (i=0; i<n; i++)
    {
    water += min(left[i], right[i]) - arr[i];

    }
```


3 2 0 4 6
3
2
4
6

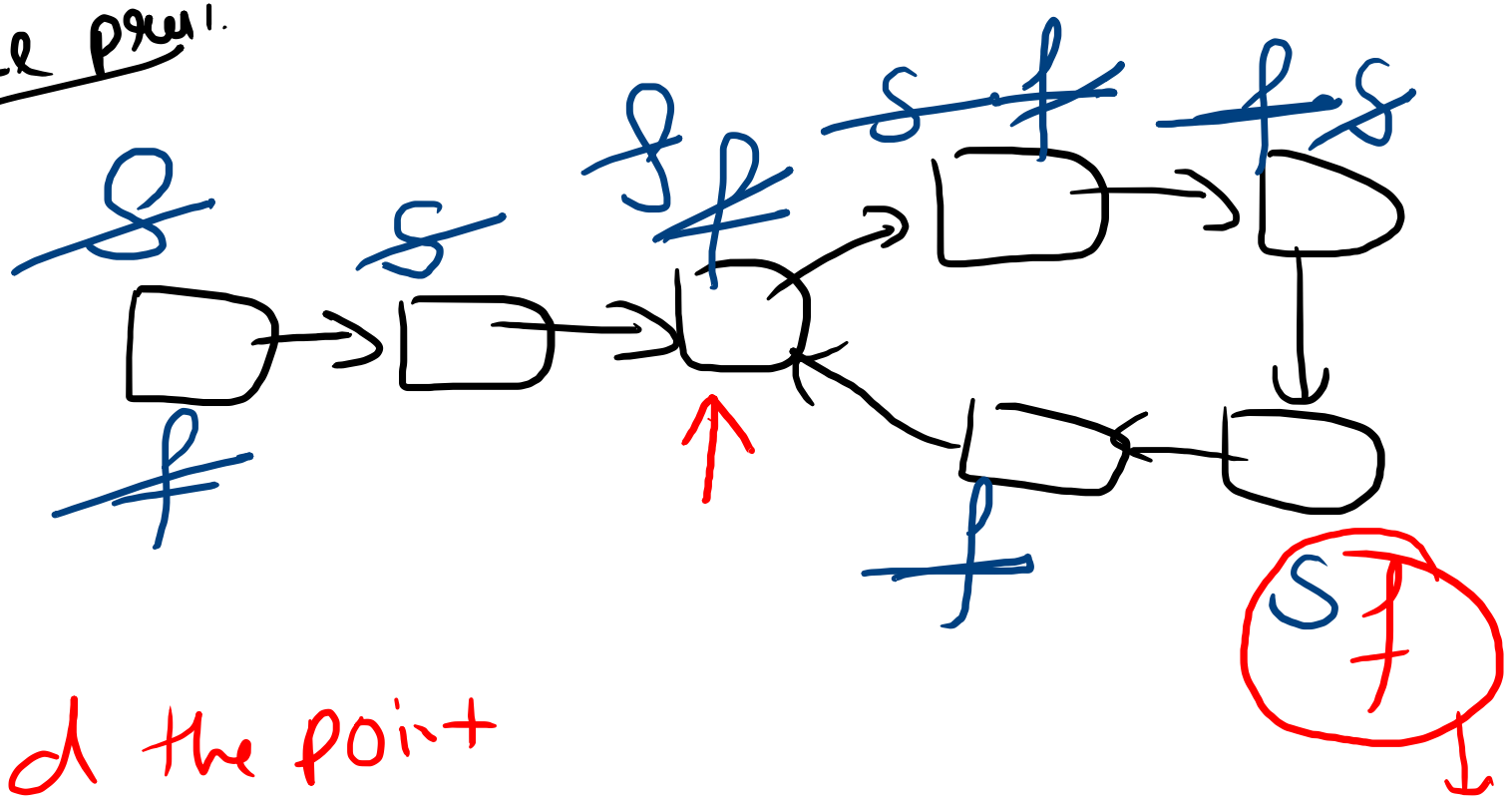

first
Next Smaller
element
first
Next Smaller
element
on
left Side
on
right.

Maximum rectangle area in histogram.

2 1 5 6 2 3



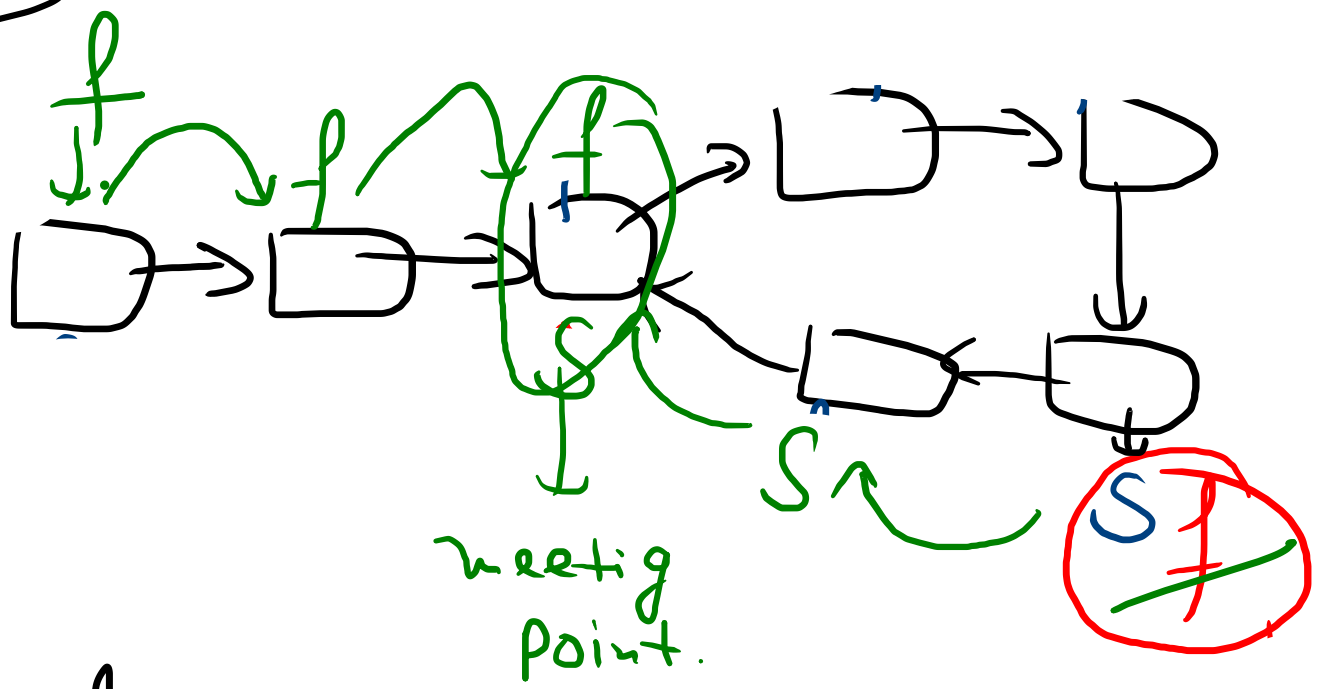
Cycle prev.



find the point where cycle is starting.

Cycle present.

Cycle prob.



slow = slow.next

fast = fast.next.next

Constra = 100 nodes

4-5

Insert
Delete.
Cycle.
Reverse.
Merge