Saloni Gupta

22BCS16659

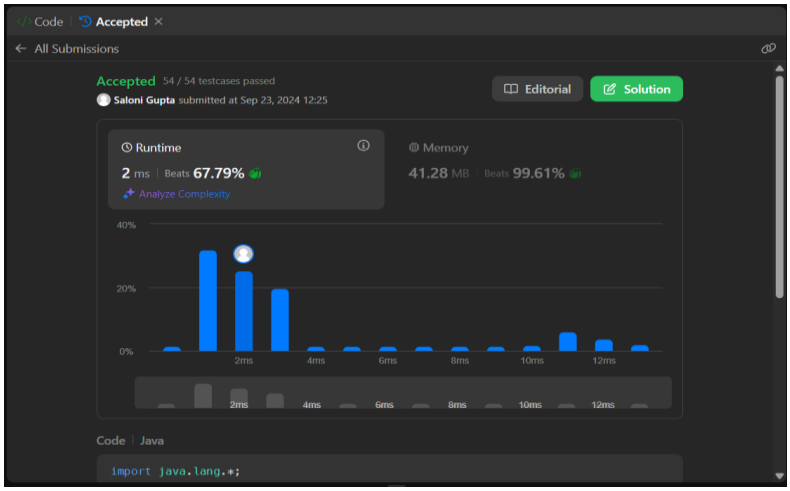
FL_IOT-603/A

Worksheet 6

53. Maximum Subarray

Code:

```
class Solution {
    public int maxSubArray(int[] nums) {
        int subArr = 0;
        int maxSub = Integer.MIN_VALUE;
        for(int i = 0; i<nums.length; i++){
            subArr = Math.max(nums[i] , subArr + nums[i]);
            maxSub = Math.max(subArr, maxSub);
        }
        return maxSub;
    }
}
```



218. The Skyline Problem

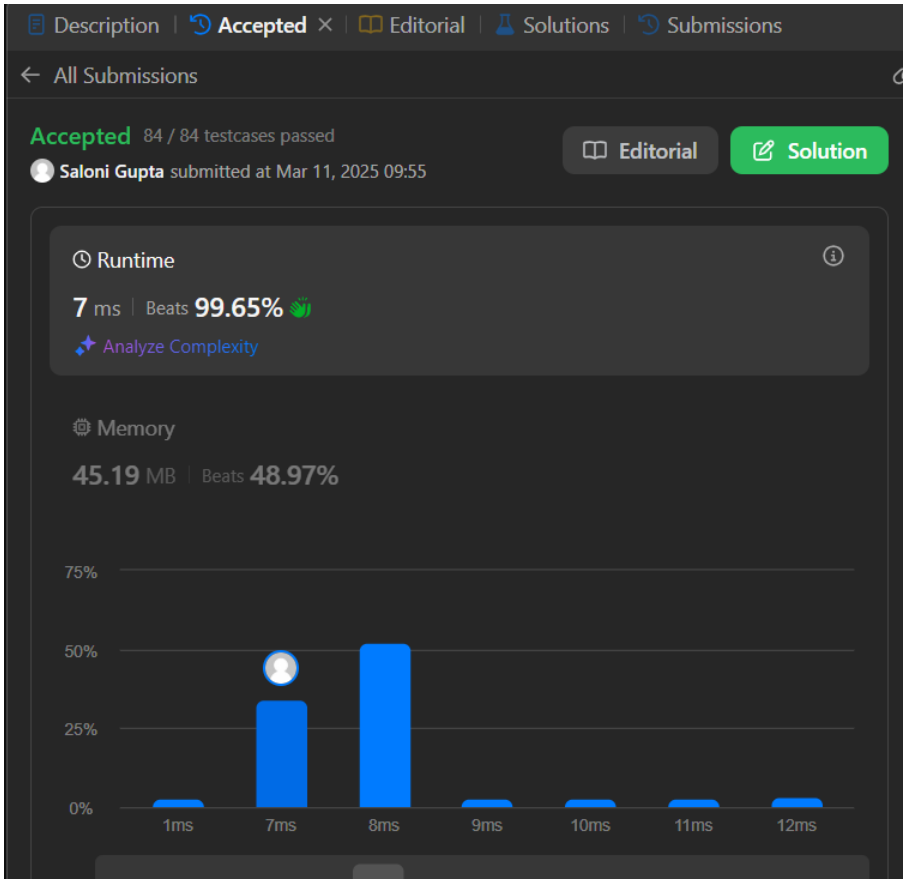
Code:

```
class TopNode {
    int x;
    int h;
    TopNode next;
    TopNode() {
    }
    TopNode(int x, int h) {
        this.x = x;
        this.h = h;
    }

    void insert(TopNode n) {
        n.next = next;
        next = n;
    }

}
class Solution {
    static final int LEFT=0, RIGHT=1, HEIGHT=2;
    public List<List<Integer>> getSkyline(int[][] buildings) {
        TopNode head = new TopNode(0,0);
        head.insert(new TopNode(Integer.MAX_VALUE, 0));
        TopNode start = head;

        for (int i = 0; i<buildings.length; i++) {
            int[] b = buildings[i];
            int bL = buildings[i][LEFT];
```

```
int bR = buildings[i][RIGHT];
int bH = buildings[i][HEIGHT];
//System.out.println(Arrays.toString(buildings[i]));
while (bL >= start.next.x) { start = start.next; }
//System.out.println(start.toString());
for (TopNode t = start ; bR > t.x; t = t.next) {
   //System.out.println(head.toString());
   if (bH <= t.h) {
      continue;
   }

   TopNode stop = t;
   while (stop.next != null && stop.next.x < bR && stop.next.h <= bH ) {
      stop = stop.next;
   }

   if (bL <= t.x) {
      if (bR >= stop.next.x) {
         t.next = stop.next;
         t.h = bH;
      }
      else if (t == stop) {
         t.insert(new TopNode(bR,t.h));
         t.h = bH;
         break;
      }
      else {
         stop.x = bR;
```

```
            t.h = bH;
            t.next = stop;
            break;
        }
    }
    else {
        if (bR >= stop.next.x) {
            if (t == stop) {
                t.insert(new TopNode(bL, bH));
            }
            else {
                t.next = stop;
                stop.x = bL;
                stop.h = bH;
            }
            break;
        }
        else if (t == stop) {
            t.insert(new TopNode(bL,bH));
            t.next.insert(new TopNode(bR,t.h));
            break;
        }
        else {
            t.next = stop;
            t.insert(new TopNode(bL,bH));
            stop.x = bR;
            break;
        }
```

```
            }
            t = stop;

        }
    }

    List<List<Integer>> skyline = new ArrayList<>();

    if (head.h == 0)
        head = head.next;
    while (head != null) {
        int height = head.h;
        skyline.add(List.of(head.x, height));
        while ( (head = head.next) != null && head.h == height) {}
    }

    return skyline;
  }

}
```

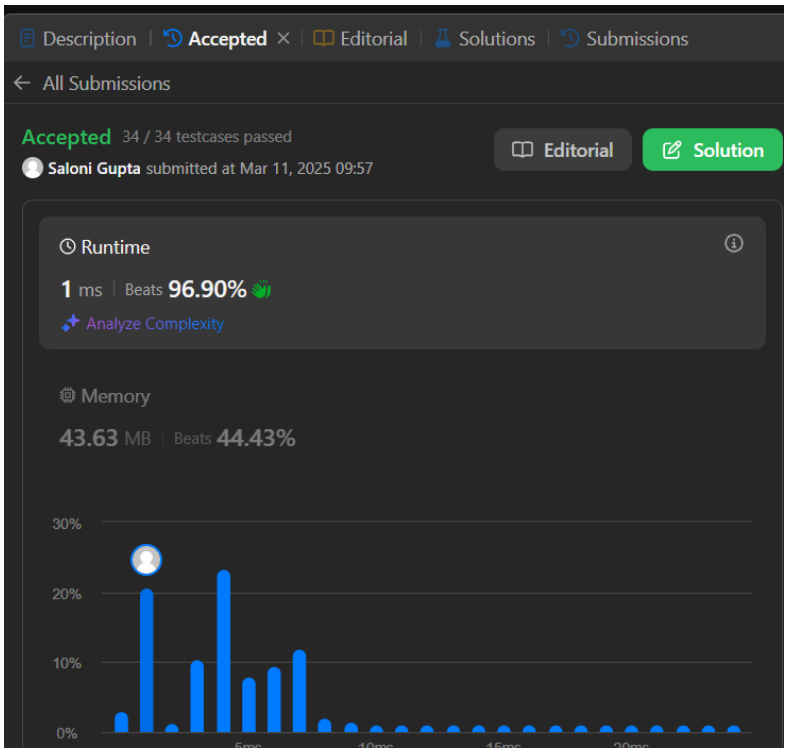


108. Convert Sorted Array to Binary Search Tree

Code:

```
class Solution {
    public TreeNode sortedArrayToBST(int[] nums) {
        return helper(nums, 0, nums.length - 1);
    }

    private TreeNode helper(int[] nums, int left, int right) {
        if (left > right) return null;
        int mid = (left + right) / 2;
        TreeNode root = new TreeNode(nums[mid]);
        root.left = helper(nums, left, mid - 1);
        root.right = helper(nums, mid + 1, right);
        return root;
```

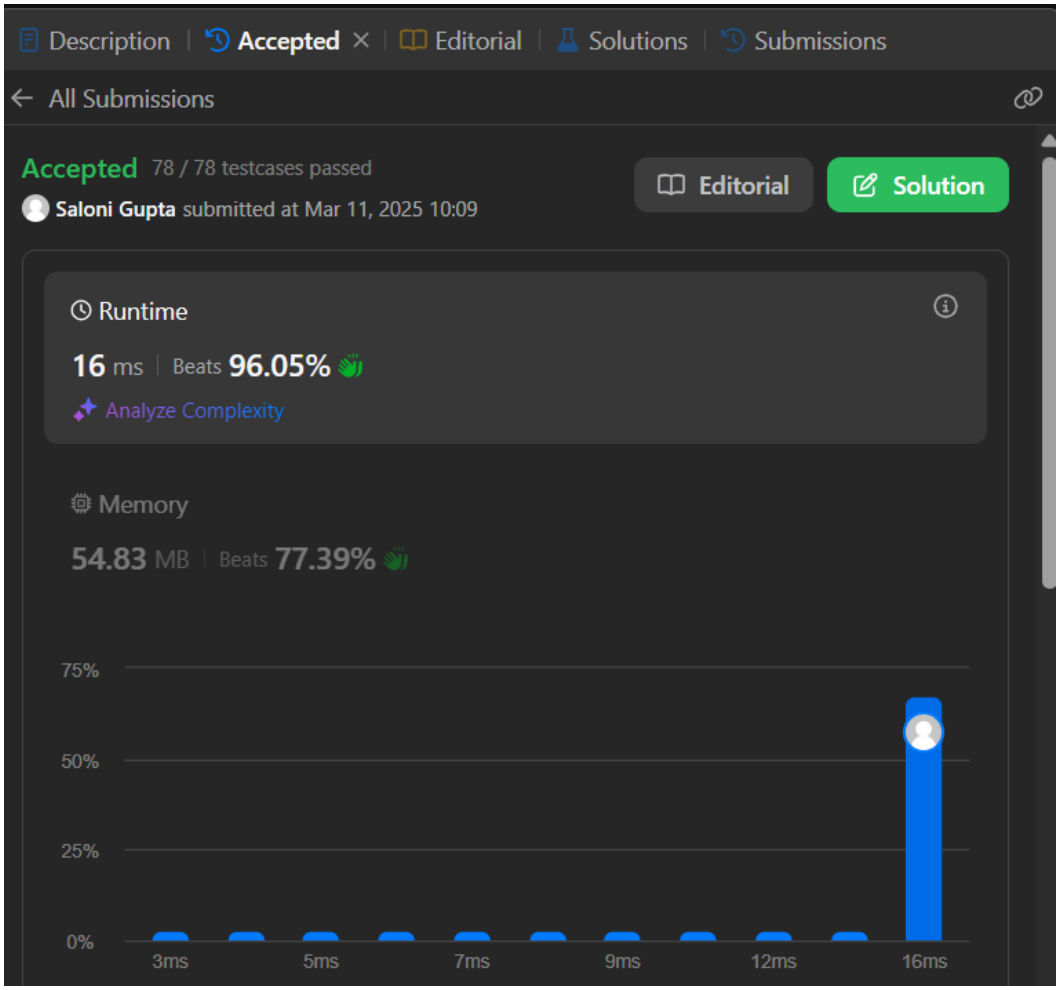
```
    }
}
```



191. Number of 1 Bits

Code:

```
class Solution {
    public int hammingWeight(int n) {
        int count = 0;
        while (n > 0) {
            n = n & (n - 1); // Clears the lowest set bit
            count++;
        }
        return count;
    }
}
```


Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 78 / 78 testcases passed
Saloni Gupta submitted at Mar 11, 2025 10:09
Editorial
Solution
Runtime
16 ms | Beats 96.05%
Analyze Complexity
Memory
54.83 MB | Beats 77.39%
75%
50%
25%
0%
3ms
5ms
7ms
9ms
12ms
16ms