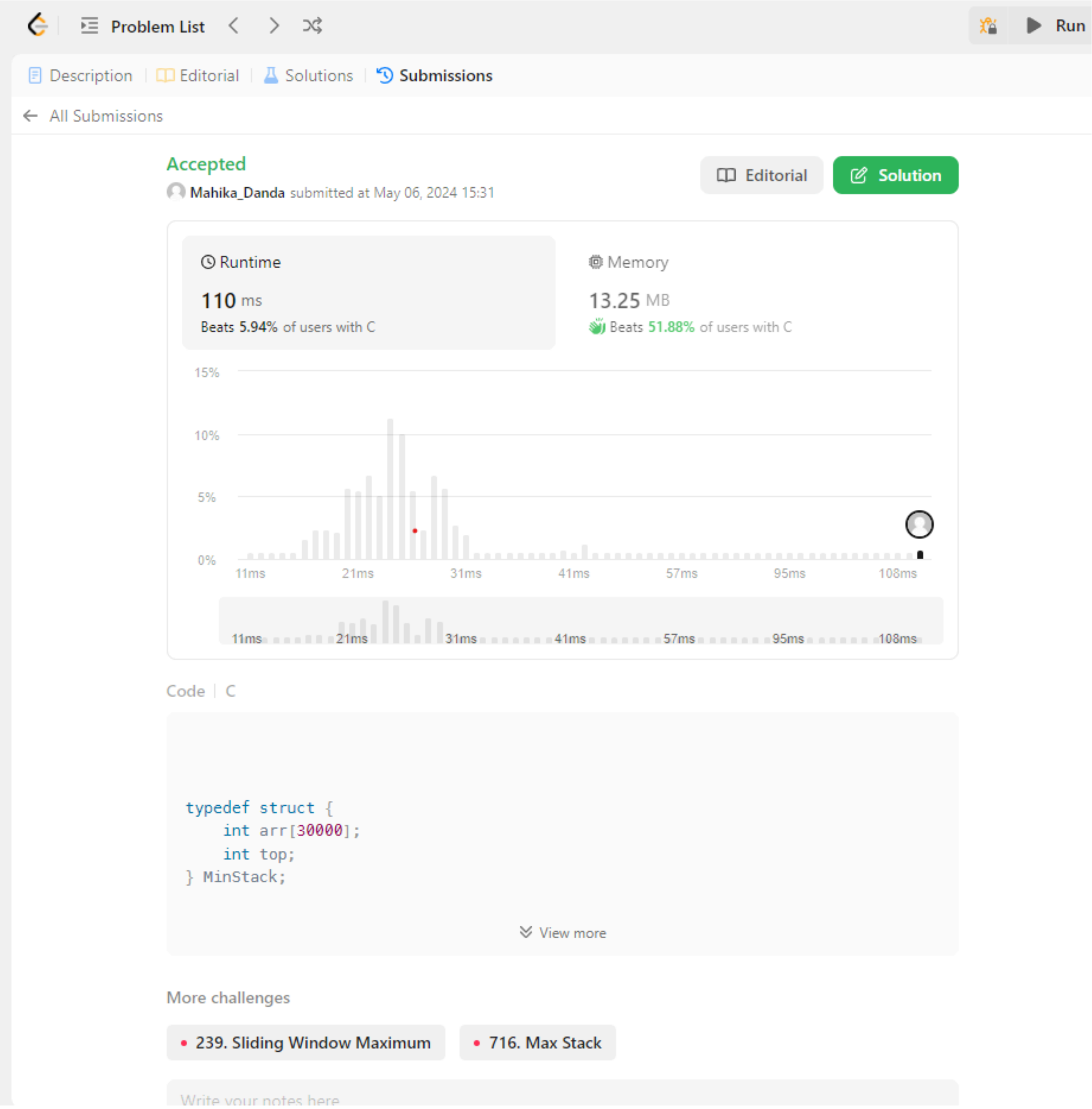
**Q) Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.**

```
typedef struct {
    int arr[30000];
    int top;
} MinStack;


MinStack* minStackCreate() {
    MinStack *minStack = (MinStack*)malloc(sizeof(MinStack));
    minStack->top=-1;
    return minStack;
}

void minStackPush(MinStack* obj, int val) {
    /*if(obj==NULL){
        obj->min=val;
    }
    else {
        if(val<obj->min){
        obj->min=val;
        }
    }*/
    obj->arr[++obj->top]=val;
}

void minStackPop(MinStack* obj) {
    obj->top--;

}

int minStackTop(MinStack* obj) {
    return obj->arr[obj->top];
}

int minStackGetMin(MinStack* obj) {
    int min=obj->arr[0];
    for(int i=0;i<=obj->top;i++){
        if(obj->arr[i]<min){
            min=obj->arr[i];
        }
    }
    return min;
}

void minStackFree(MinStack* obj) {
    free(obj);
}
```



Accepted

Mahika_Danda submitted at May 06, 2024 15:31

Runtime

110 ms

Beats 5.94% of users with C

Memory

13.25 MB

Beats 51.88% of users with C

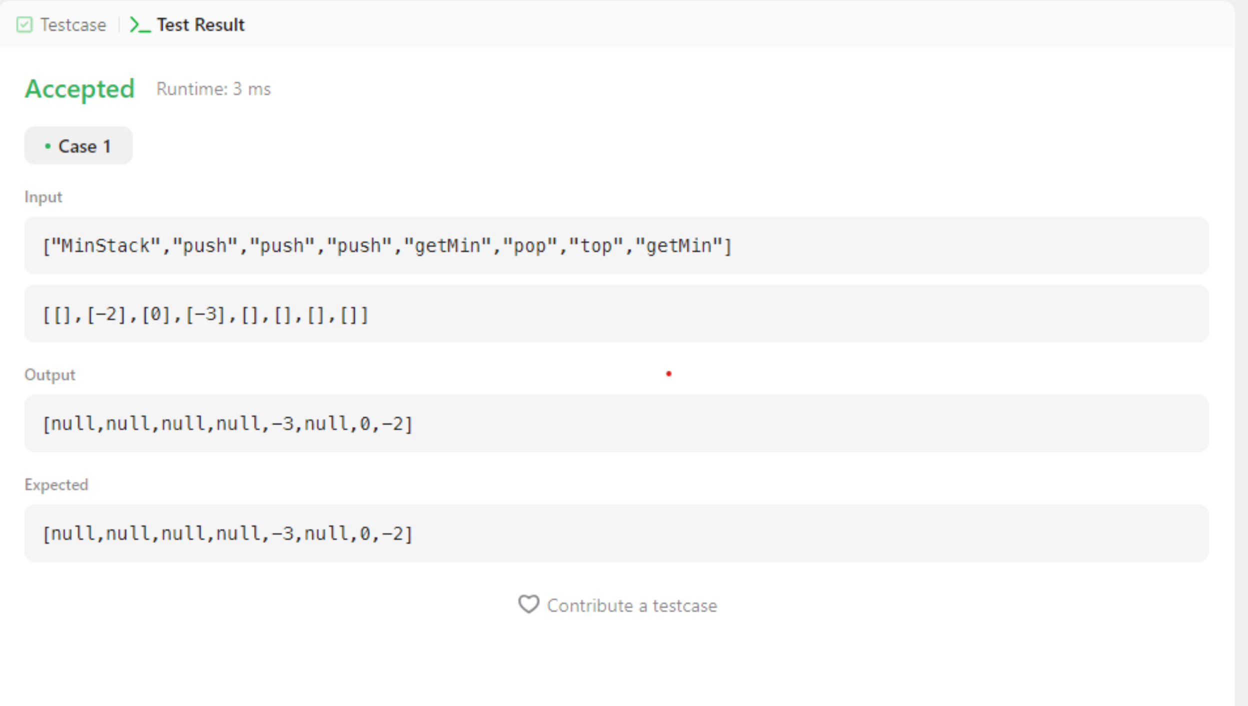
Code | C

```
typedef struct {
    int arr[30000];
    int top;
} MinStack;
```

More challenges

- 239. Sliding Window Maximum
- 716. Max Stack

Write your notes here



Accepted Runtime: 3 ms

• Case 1

Input

```
["MinStack","push","push","push","getMin","pop","top","getMin"]
```

```
[[],[-2],[0],[-3],[],[],[],[]]
```

Output

```
[null,null,null,null,-3,null,0,-2]
```

Expected

```
[null,null,null,null,-3,null,0,-2]
```