# Min Stack

Using an extra stack

Space Complexity : O(N)

hm ek external stack ka use krenge to maintain min elem. Jb bhi stack me push krna ho to orig wali me to hmesa hi push hoga lkn min stack me tbhi push hoga jb wo empty ho which means ki pehla elem hi min h to usko push krdiya aur min stack me value tb bhi push hogi jb min stack k top ka elem greater than curr elem h.

Coming to pop

orig stack se pop as usual lkn min stack se tbhi pop hoga jb dono stack ka top elem same ho this means ki agr min elem hi stack se pop hora ho to usko min stack me store krne ka koi use ni h also agr duplicate min values h to wo do bar min stack me push hongi kyunki agr ek min nikal bhi jaye to bhi 1 min rahe stack me jo rest of elem k liye abhi bhi min value h.

top me simply return top elem and get min me min stack ka top elem return krdo.

```
class MinStack {
public:
    stack<int> s;
    stack<int> ss;
    MinStack()
    {

    }
    void push(int val) {
        s.push(val);
        if(ss.empty())
        {
            ss.push(val);
        }
        else if(ss.top()>=val)
        {
            ss.push(val);
        }
    }

    void pop() {
        if(ss.top()==s.top())
        {
            s.pop();
```

```
            ss.pop();
        }
        else
        {
            s.pop();
        }
    }

    int top() {
       return s.top();
    }

    int getMin() {
        return ss.top();
    }
};

/**
 * Your MinStack object will be instantiated and called as such:
 * MinStack* obj = new MinStack();
 * obj->push(val);
 * obj->pop();
 * int param_3 = obj->top();
 * int param_4 = obj->getMin();
 */
```