

## Design a Min Stack

→ Enabling the `getmin()` function

### Pseudocode

#### 1st Approach

```
class MinStack {
    Stack<pair<int, int>> s;
    MinStack() {}
    void push(int val) {
        if (s.empty()) {
            s.push(val, val);
        } else {
            minval = min(val, s.top().second);
            s.push(val, minval);
        }
    }
    void pop() {
        s.pop();
    }
    int top() {
        return s.top().first;
    }
    int getmin() {
        return s.top().second;
    }
}
```

push  
pop  
top  
← getmin  
Enable it using }

void pop() {  
s.pop();  
}

top() {

return s.top().first;

getmin()

return s.top().second;

^ push of

$s.c \sim (2 \times n)$



## 2nd Approach

```
getMin() {  
    return minval; }  
push(val) {  
    if (s.empty())  
        s.push(val)  
        minval = val  
    if (val < minval) {  
        s.push(2 * val - minval)  
        minval = val  
    }  
    else {  
        s.push(val)  
    }  
}  
pop() {  
    if (s.top() < minval) {  
        minval = 2 * minval - s.top();  
        s.pop();  
    }  
    s.pop();  
}  
top() {  
    s.top() — minval — return minval  
    else  
        return s.top()  
}  
[S.C = O(n)]
```