

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

Implement the `MinStack` class:

- `MinStack()` initializes the stack object.
- `void push(int val)` pushes the element `val` onto the stack.
- `void pop()` removes the element on the top of the stack.
- `int top()` gets the top element of the stack.
- `int getMin()` retrieves the minimum element in the stack.

You must implement a solution with  $O(1)$  time complexity for each function.

### Solution:

```
class MinStack {
    stack<int> V;
    stack<int> M;
public:
    MinStack() {

    }

    void push(int val) {
        V.push(val);
        if (M.empty())
            M.push(val);
        else
            M.push(min(M.top(), val));
    }

    void pop() {
        V.pop(), M.pop();
    }

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

    int getMin() {
        return M.top();
    }
};
```