LeetCode 155 Min Stack

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

- push(x) – Push element x onto stack.
- pop() – Removes the element on top of the stack.
- top() – Get the top element.
- getMin() – Retrieve the minimum element in the stack.

Example

```
MinStack minStack = new MinStack();
minStack.push(-2);
minStack.push(0);
minStack.push(-3);
minStack.getMin();   --> Returns -3.
minStack.pop();
minStack.top();      --> Returns 0.
minStack.getMin();   --> Returns -2.
```

Method

The difference of the min stack with normal stack is the getMin() function. At the first glance, I think I should have a variable to store the minimum. But the problem is if we pop() the minimum, we don’t know the next minimum. So I think we need to save all minimum. At first I choose an

array to store them. This solution is feasible, but why don't I use the stack class which has pop(), push() and top() functions? I don't need to implement this functions myself.

```
class MinStack {
public:
    /** initialize your data structure here. */
    stack<int> num, min;
    MinStack() {
    }
    void push(int x) {
        num.push(x);
        if(min.empty()==true || x<=min.top()){
            min.push(x);
        }
    }
    void pop() {
        if(num.top() == min.top())
            min.pop();
        num.pop();
    }
    int top() {
        return num.top();
    }
    int getMin() {
        return min.top();
    }
};
```

Note

Actually we don’t need to store all minimum in every push(), we only store the new mininum. But we should pay more attention to a curve case. When we push(min), we also need to store a same min, otherwise, after pop(), we can’t find min.

Follow up:

If we keep saving duplicate number, space will approximately equal to O(n), how to decrease space?

We can plus count when we store the minimum.

```
class MinStack {
public:
    stack<int> num;
    stack<pair<int, int>> min;
    //int count=0; we can't use this public variable
    MinStack() {};
    void push(int x) {
        num.push(x);
        if(min.empty() || x < min.top().first)
            min.push(make_pair(x,1));
        else if(x == min.top().first)
            min.top().second++;
    }
```

```
    void pop() {
        if(num.top() == min.top().first){
            min.top().second--;
            if( min.top().second ==0)
                min.pop();
        }
        num.pop();
    }
    int top(){
        return num.top();
    }
    int getMin(){
        return min.top().first;
    }
};
```

Note

```
How to define a pair in C++
pair<int,int> pair1;
pair1 = make_pair(2,3);
pair<string,int> pair2("Elly", 30);
```

Summary

Cash Model

In the stack application, when we want to decrease space, we need trying

to change the content that will be saved in the stack.