

#LeetCode155

<https://leetcode.com/problems/min-stack/description/>

Yifeng Zeng

#题目描述

Min Stack

#思路报告

这个题是去年听太阁算法左程云老师的课讲的，当时没多想。后来仔细想一下其实挺straight forward的，要实现一个min stack那么stack本身的功能就用一个ArrayDeque作为stack来完成。那么min这个property肯定需要另外维护。什么数据结构最接近stack呢，肯定先想到stack本身，那么另外用一个stack（minStack）来维护min这个property可以不呢，当然可以。push的时候push当前整个stack里面最小的数就好，那么push进minStack时，peek minStack的值跟要push的x比较好，因为peek minStack已经是整个stack里面最小的值了。x小就push x，否则push minStack.peek()。

代码如下

```
class MinStack {

    Deque<Integer> dataStack;// = new ArrayDeque<>();
    Deque<Integer> minStack;// = new ArrayDeque<>();
    /** initialize your data structure here. */
    public MinStack() {
        dataStack = new ArrayDeque<>();
        minStack = new ArrayDeque<>();
    }

    public void push(int x) {
        dataStack.push(x);
        if (minStack.isEmpty() || minStack.peek() > x) {
            minStack.push(x);
        } else {
            minStack.push(minStack.peek());
        }
    }
}
```

```
public void pop() {  
    dataStack.pop();  
    minStack.pop();  
}  
  
public int top() {  
    return dataStack.peek();  
}  
  
public int getMin() {  
    return minStack.peek();  
}  
}
```

#套路总结

- 既然题目是设计min stack，那么先考虑把basic的stack功能用ArrayDeque实现，再考虑怎样维护min这个property。先考虑用跟stack有最接近性质的stack本身来维护min，那么只要保证每次push的是整个stack里面的最小值就可以了，pop的时候两个stack同时pop，getMin的时候peek minStack就好。