



链滴

[每日 LeetCode] 155. Min Stack

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1557234962274>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>原文链接 [每日 LeetCode] 155. Min Stack</p>

<p>Description:</p>

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

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.

<p>Example:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">MinStack minStack = new MinStack();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.push(-2);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.push(0);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.push(-3);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.getMin();
--&gt; Returns -3.
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.pop();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.top();
--&gt; Returns 0.
</span> </span> <span class="highlight-line"> <span class="highlight-cl">minStack.getMin();
--&gt; Returns -2.
</span> </span> </code> </pre>
```

<hr>

<blockquote>

<p>本题要求设计一个最小栈，即在原有栈的方法中加入 <code>getMin()</code> 方法。</p>

</blockquote>

<p>思路一：最开始想到的是使用 vector 来实现，最后运行起来效率太低，不推荐。</p>

<p>思路二：使用两个栈，一个正常操作，另一个栈保证栈顶元素最小，这个效率很高，也是 LeetCode 高赞答案，值得推荐。</p>

<hr>

<p>C++ 代码（思路一，不推荐）</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl">class MinStack {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">public:
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    /** initialize your
data structure here. */
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    MinStack() {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    void push(int x)
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        vec.push_back
(x);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    void pop() {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        vec.pop_back
};
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    }
</span> </span> <span class="highlight-line"> <span class="highlight-cl">    int top() {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">        return vec[ve
```

```

.size() - 1];
}
int getMin() {
    return *min_element(vec.begin(), vec.end());
}
private:
    vector<int> &g
; vec;
};
/**
 * Your MinStack
bject will be instantiated and called as such:
 * MinStack* obj =
new MinStack();
 * obj->push(x);
 * obj->pop();
 * int param_3 =
bj->top();
 * int param_4 =
bj->getMin();
 */

```

运行时间: 208ms

运行内存: 17.1M

C++ 代码 (思路二, 推荐)

```

class MinStack {
public:
    /** initialize you
data structure here. */
    MinStack() {
    }
    void push(int x)
    {
        s1.push(x);
        if (s2.empty()
| x <= getMin()) s2.push(x);
    }
    void pop() {
        if (s1.top() ==
getMin()) s2.pop();
        s1.pop();
    }
    int top() {
        return s1.top(
;
    }
    int getMin() {
        return s2.top(
;
    }
}

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">private:
</span></span><span class="highlight-line"><span class="highlight-cl">    stack<int>
s1, s2;
</span></span><span class="highlight-line"><span class="highlight-cl">};
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">/**
</span></span><span class="highlight-line"><span class="highlight-cl"> * Your MinStack
bject will be instantiated and called as such:
</span></span><span class="highlight-line"><span class="highlight-cl"> * MinStack* obj =
new MinStack();
</span></span><span class="highlight-line"><span class="highlight-cl"> * obj->push(x);
</span></span><span class="highlight-line"><span class="highlight-cl"> * obj->pop();
</span></span><span class="highlight-line"><span class="highlight-cl"> * int param_3 =
bj->top();
</span></span><span class="highlight-line"><span class="highlight-cl"> * int param_4 =
bj->getMin();
</span></span><span class="highlight-line"><span class="highlight-cl"> */
</span></span></code></pre>
<p>运行时间: 32ms</p>
<p>运行内存: 17M</p>

```