**难度: 简单**

请设计一个栈，除了常规栈支持的pop与push函数以外，还支持min函数，该函数返回栈元素中的最小值。执行push、pop和min操作的时间复杂度必须为O(1)。

示例：

MinStack minStack = new MinStack();

minStack.push(-2);

minStack.push(0);

minStack.push(-3);

minStack.getMin(); --> 返回 -3.

minStack.pop();

minStack.top(); --> 返回 0.

minStack.getMin(); --> 返回 -2.

通过次数10,728提交次数17,792

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/min-stack-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1(使用双指针链表):**

**由于**getMin()方法使用了循环导致速度相对较慢

|  |
| --- |
| public class MinStack {  private Node head; //头结点  private Node end; //尾结点  public MinStack() {  head = new Node();  end = head;  }  public void push(int x) {  Node node = new Node();  node.value = x;  //指向该结点  end.next = node;  //指向上一个结点  node.prior = end;  end = node;  }  public void pop() {  //最后个结点给到temp  Node temp = end;  //把end的上一个结点给到end  end = temp.prior;  //end的上一个结点指向null  temp.prior.next = null;  }  public int top() {  return end.value;  }  public int getMin() {  Node temp = head.next;  int min = temp.value;  while (temp.next != null) {  if (min > temp.next.value) {  min = temp.next.value;  }  temp = temp.next;  }  return min;  }  class Node {  public Node prior; //前指针  public int value;  public Node next; //后指针  } |