# [最小栈](https://leetcode-cn.com/explore/interview/card/bytedance/245/data-structure/1049/)

**头条重点**

## 题目

设计一个支持 push，pop，top 操作，并能在常数时间内检索到最小元素的栈。

push(x) -- 将元素 x 推入栈中。 pop() -- 删除栈顶的元素。 top() -- 获取栈顶元素。 getMin() -- 检索栈中的最小元素。

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.

## 解题思路

class MinStack {  
  
 /\*\* initialize your data structure here. \*/  
 public MinStack() {  
  
 }  
  
 private LinkedList<Integer> stack = new LinkedList<>();  
 private Queue<Integer> minStack = new PriorityQueue<>();  
  
  
  
 public void push(int x) {  
 stack.offerFirst(x);  
 minStack.offer(x);  
 }  
  
 public void pop() {  
 Integer poll = stack.pollFirst();  
 if (poll != null) {  
 minStack.remove(poll);  
 }  
 }  
  
 public int top() {  
 Integer first = stack.peekFirst();  
 return first == null ? 0 : first;  
 }  
  
 public int getMin() {  
 Integer first = minStack.peek();  
 return first == null ? 0 : first;  
 }  
}