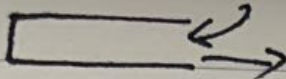


栈.  列表. append. pop 实现.  
peek 用 list[-1] 实现.

假设每个时刻调用 getMin 即可得到栈的最小值,  
最简单的思路即: 把栈全数倒出来取 Min 即可  
时:  $O(n)$  空间:  $O(1)$  但破坏了栈结构. 既然 ST, 就需要  
额外空间来记录  $O(n)$

换个思路: 每次栈的更新, 都有内部 min 的更新.  
时:  $O(1)$  空间:  $O(1)$

Class MyStack:

```
def __init__(self):
```

```
    self.stack = []
```

```
    self.min = self.stack []
```

```
def append(self, item):
```

```
    self.stack.append(item)
```

```
def pop(self, item):
```

```
    self.stack.pop(item)
```

```
def peek:
```

```
    self.min.pop()
```

怎么办? 给个记忆.

[5, 4, 2, 2, 1, 0, 0]

```
def getMin(self, item):
```

```
    return self.min[-1]
```

时:  $O(1)$  空:  $O(n)$