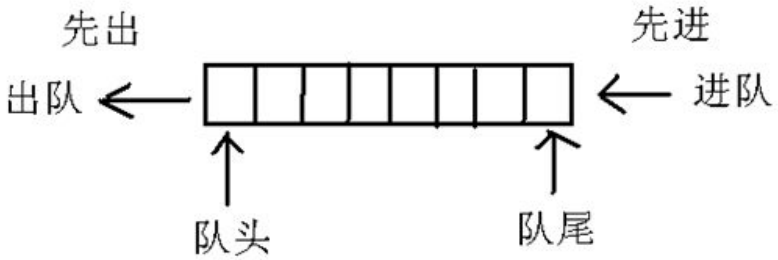
一． 队列数据结构的封装

队列类。队列(queue)是具有先进先出(FIFO)特性的数据结构。一个队列就像是一行队伍,数据从前端被移除,从后端被加入。这个类必须支持下面几种方法:



并实现下面的功能:

Operation	Return Value	first ← Q ← last
Q.enqueue(5)	–	[5]
Q.enqueue(3)	–	[5, 3]
len(Q)	2	[5, 3]
Q.dequeue( )	5	[3]
Q.is_empty( )	False	[3]
Q.dequeue( )	3	[ ]
Q.is_empty( )	True	[ ]
Q.dequeue( )	“error”	[ ]
Q.enqueue(7)	–	[7]
Q.enqueue(9)	–	[7, 9]
Q.first( )	7	[7, 9]
Q.enqueue(4)	–	[7, 9, 4]
len(Q)	3	[7, 9, 4]
Q.dequeue( )	7	[9, 4]

注意： PPT 上的参考代码不一定是最有答案

二. 最近请求次数(队列的应用)

写一个 `RecentCounter` 类来计算最近的请求。

它只有一个方法：`ping(int t)` ，其中 `t` 代表以毫秒为单位的某个时间。

返回从 3000 毫秒前到现在的 `ping` 数。

任何处于 `[t - 3000, t]` 时间范围之内的 `ping` 都将会被计算在内，包括当前（指 `t` 时刻）的 `ping`。

保证每次对 `ping` 的调用都使用比之前更大的 `t` 值。

示例：

输入：inputs = ["RecentCounter","ping","ping","ping","ping"], inputs = [[],[1],[100],[3001],[3002]]
输出：[null,1,2,3,3]

提示：

1. 每个测试用例最多调用 `10000` 次 `ping`。
2. 每个测试用例会使用严格递增的 `t` 值来调用 `ping`。
3. 每次调用 `ping` 都有 `1 <= t <= 10^9`。

代码格式:

```
class RecentCounter:
    def __init__(self):
        pass
    def ping(self, t: int) -> int:
        pass
# Your RecentCounter object will be instantiated and called as such:
# obj = RecentCounter()
# param_1 = obj.ping(t)
```

三. 常见 python 面试题目整理

1. 列举 Python2 和 Python3 的区别？

2. 简述 Python 的深浅拷贝以及应用场景？

3. 能否解释一下 *args 和 **kwargs?

4. 简述 生成器、迭代器、可迭代对象 以及应用场景？

5. 请说明 yield 关键字的工作机制。

6. 请简单谈谈装饰器的作用和功能。

7. Python 中如何读取大数据的文件内容?

8. Python 中的模块和包是什么?

9. python 是如何进行内存管理的(python 是如何实现垃圾回收机制的)?

10. 谈谈你对面向对象的理解？

11. Python 面向对象中的继承有什么特点？

12. 面向对象中 super 的作用？

13. 面向对象深度优先和广度优先是什么，并说明应用场景？

14. 请简述__init__和__len__这两个魔术方法的作用