

## 数组链表；堆栈队列

1. 数组链表

2.

### 反转链表 - reverse linked list

```
4
5 def reverseList(self, head):
6     cur, prev = head, None
7     while cur:
8         cur.next, prev, cur = prev, cur, cur.next
9     return prev
10
```

### 链表交换相邻元素 - swap node

```
2
3 def swapPairs(self, head):
4     pre, pre.next = self, head
5     while pre.next and pre.next.next:
6         a = pre.next
7         b = a.next
8         pre.next, b.next, a.next = b, a, b.next
9         pre = a
10    return self.next
11
```

## 探测环 - detect cycle



```
2
3 def hasCycle(self, head):
4     fast = slow = head
5     while slow and fast and fast.next:
6         slow = slow.next
7         fast = fast.next.next
8     if slow is fast:
9         return True
10    return False
11
```

### 2. 堆栈

## 20. Valid Parentheses

```
def isValid(self, s):
    stack = []
    paren_map = {')': '(', ']': '[', '}': '{'}
    for c in s:
        if c not in paren_map:
            stack.append(c)
        elif not stack or paren_map[c] != stack.pop():
            return False
    return not stack
```

用栈实现队列

```

    * 定义两个栈来实现队列
    * 栈A 负责插入新元素
    * 栈B 负责移除老元素
    */
private Stack<Integer> stackA = new Stack<>();
private Stack<Integer> stackB = new Stack<>();

/**
 * 入队操作
 * @Param element
 */
public void enqueue(int element){
    stackA.push(element);
}

/**
 *
 * 出队操作
 */
public Integer dequeue(){
    if (stackB.isEmpty()){
        if (stackA.isEmpty()){
            return null;
        }
        fetchFormStackA();
    }

    return stackB.pop();
}

/**
 * 从stackA栈中拿到出栈元素压入栈B
 */
private void fetchFormStackA() {
    while (!stackA.isEmpty()){
        stackB.push(stackA.pop());
    }
}
}

```

## 2.pop() -- 移除栈顶元素

栈的栈顶是最后一个元素，按照栈的规则是FILO（先进后出），而栈的栈顶对于队列来说就是最后一个要出来的。

要移除栈顶元素就要先将队列前面的值全部移走再进去，就完成了POP的操作

```
1 int pop() {
2     int size = q.size()-1;
3     for(int i=0;i<size;i++)
4     {
5         int data = q.front();    //取出口第一个数保存
6         q.pop();                //取出这个数
7         q.push(data);           //将取出的数放到队列的最后
8     }
9     int d = q.front();          //当把所有前面的值都移走了之后，再把第一个元素移除
10    q.pop();
11    return d;
12
13 }
```