

# 两个栈实现一个队列

栈有入栈和出栈，要实现队列，也就是实现队列入队和出队的功能。队列是先入队先出队，和栈不同，栈是先入栈后出栈。

队列是逻辑结构，数组是数物理结构，无法在一起比较，没有关系，但可以用数组来表示队列。

## 解题思路

### 1. 第一个栈，先入栈

stack1.add(100), stack1.add(200), stack1.add(300), 变成了 [100, 200, 300]。

### 2. 第二个栈，入栈

从第一个栈出栈再入第二个栈，依次循环执行第一个栈长度的次数，第二个栈变成 [300, 200, 100]，就是第一个栈的反序。

### 3. 第二个栈出栈

第二个栈出栈一次，把最后一个元素删掉。变成了 [300, 200]。

### 4. 第一个栈再次入栈

将第二个栈出栈入第一个栈，依次循环执行第二个栈长度的次数，第一个栈变成 [200, 300]。

经过以上4步，达到了通过两个栈来实现先入先出的目标，这就是队列的功能。

## 代码实现

```
// 两个栈实现一个队列
class twoStackOneQueue {
  constructor(stack1, stack2) {
    this.stack1 = stack1;
    this.stack2 = stack2;
  }
  // O(1)
  add(item) {
    this.stack1.push(item);
  }
  // O(n)
  delete() {
    let res = [];
    while(this.stack1.length) {
      const item = this.stack1.pop();
      this.stack2.push(item);
    }
    res = this.stack2.pop();
    while(this.stack2.length) {
      const item = this.stack2.pop();
      this.stack1.push(item);
    }
  }
}
```

```
        }
        return res;
    }
    get stack() {
        return this.stack1;
    }
    get length() {
        return this.stack1.length;
    }
}

let queue = new twoStackOneQueue([], []);
queue.add(100);
queue.add(200);
queue.add(300);
console.log(queue.length);
console.log(queue.stack);
console.log(queue.delete());
console.log(queue.stack);
console.log(queue.length);
```