

班级_____

实验题目：栈和队列的应用关键算法实现及性能分析

一、实验目的

1. 能够运用高级程序设计技术实现栈和队列的应用及其关键算法
2. 理解栈和队列的工作原理

二、实验原理

1. 栈和队列的基本原理

(1)、栈的原理

后进先出，栈中弹出的元素总是你最后放进去的那个类似于数组，只是限制了存取操作的灵活性。

(2)、队列的原理

队列只允许在表的前端（front）进行删除操作，而在表的后端（rear）进行插入操作，和栈一样，队列是一种操作受限制的线性表。进行插入操作的端称为队尾，进行删除操作的端称为队头。

2. 关键算法设计原理

(1)、栈的实现原理

- 压栈是将元素压入数组尾部在数组的头部插入元素是一个很耗时的操作，它的时间复杂度为 $O(n)$ ，因为需要将现有元素往后移位为新元素腾出空间。
- 在尾部插入元素的时间复杂度为 $O(1)$ 栈只允许使用者从栈顶压入（push）元素，从栈顶弹出（pop）元素，取得（peek）栈顶元素但不弹出。

(2)、队列的实现原理

- 初始化队列：Init_Queue(q)，初始条件：队 q 不存在。操作结果：构造了一个空队；
- 入队操作：In_Queue(q, x)，初始条件：队 q 存在。操作结果：对已存在的队列 q，插入一个元素 x 到队尾，队发生变化；
- 出队操作：Out_Queue(q, x)，初始条件：队 q 存在且非空，操作结果：删除队首元素，并返回其值，队发生变化

三、实验方案设计

1. 存储方案设计

(1)、栈：

- 栈的顺序存储结构简称顺序栈，它是运算受限制的顺序表。顺序栈的存储结构是：利用一组地址连续的存储单元依次存放自栈底到栈顶的数据元素，同时附设指针 top 只是栈顶元素在顺序表中的位置。
- 栈的链式存储结构简称链栈。链栈是一种特殊的线性表，链栈可以通过链式线性表来实现。

(2)、队列

- 数组实现：队列可以用数组 $Q[1\cdots m]$ 来存储，数组的上界 m 即是队列所容许的最大容量。在队列的运算中需设两个指针： $head$ ，队头指针，指向实际队头元素； $tail$ ，队尾指针，指向实际队尾元素的下一个位置。一般情况下，两个指针的初值设为 0，这时队列为空，没有元素。
- 链表实现：队列采用的 FIFO(first in first out)，新元素（等待进入队列的元素）总是被插入到链表的尾部，而读取的时候总是从链表的头部开始读取。每次读取一个元素，释放一个元素。所谓的动态创建，动态释放。

2. 算法的设计和实现

(1)、栈

```
#define MaxSize 100    /* 栈最大容量 */
int top;               /* 栈顶指针 */
int mystack[MaxSize]; /* 顺序栈 */
/*判栈是否为空，空返回 true，非空返回 false */
bool isEmpty();
/* 元素 x 入栈 */
void Push(int x);
/* 取栈顶元素 */
int getTop();
/* 删除栈顶元素 */
void Pop();
```

(2)、队列

```
int QueueLen(SqQueue Q); //队列长度
```

```
int EnQueue(SqQueue &Q, Person e); //加入队列
```

```
int QueueEmpty(SqQueue &Q); //队列是否为空
```

```
int DeQueue(SqQueue &Q, Person &e); //出队列
```

3. 测试方案

(1)、栈：

测试用例：任意 int 型整数

期望输出：该整数的二进制

(2)、队列：

测试用例：伴舞问题各个人数

期望输出：配对规则及剩余情况