程序设计实践（一）

哈尔滨工业大学 计算机学院
任课教师：孙大烈教授
助教：付万增

第七讲

基础数据结构

一、指针和链表
二、堆栈和队列
三、STL
四、实战演练

数据结构

数据结构是计算机存储、组织数据的方式。
数据结构是指相互之间存在一种或多种特定关系的数据元素的集合。通常情况下，精心选择的数据结构可以带来更高的运行或者存储效率。数据结构往往同高效的检索算法和索引技术有关。

分类：线性结构（如数组），树形结构，图形结构。

指针

- 指针（Pointer）是编程语言中的一个对象，利用地址，它的值直接指向（points to）存在电脑存储器中另一个地方的值。指针，变量和数据的关系，指针就像门牌号码，变量则为房间，数据则是住户。
- 指针的声明：int* a;
- 指针的使用：int b； int* a； a = &b; *a = 3; a = new int;
- 注意：‘&’在这里是取地址符号，表示取得变量a的地址。
- 对于整形指针a， printf(“%d”,a); 和 printf(“%d”,*a); 分别输出a指向的地址和a指向地址处的值。

重写swap

问题描述：编写实现交换两个整数的子程序。

Input
3 4
Output
4 3

```c
#include<stdio.h>
void swap(int* a,int *b)
{
    int t = *b;
    *b = *a;
    *a = t;
}
int main()
{
    int x,y;
    scanf("%d%d",&x,&y);
    swap(&x,&y);
    printf("%d %d\n",x,y);
    return 0;
}
```

重写快速排序

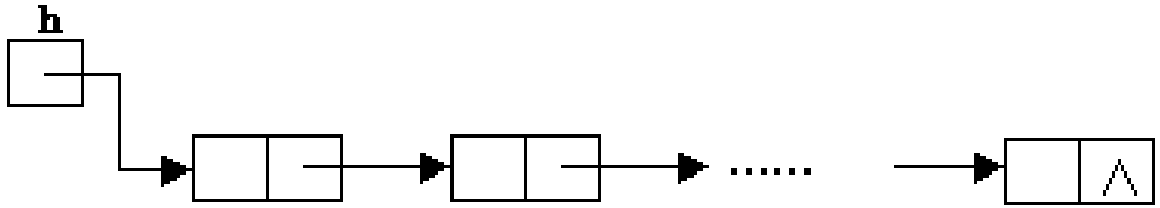
问题描述：编写实现n个数字排序的子程序。

Input
5
5 1 3 4 2
Output
1 2 3 4 5

```
int a[N];
void qsort(int* a,int l,int r)
{
    int i = l,j = r,mid = a[l+rand()%(r-l)];
    while(i<=j)
    {
        while(a[i] < mid) i++;
        while(a[j] > mid) j--;
        if(i<=j)
        {
            int t = a[i];
            a[i] = a[j];
            a[j] = t;
            i++;
            j--;
        }
    }
    if(i<r) qsort(a,i,r);
    if(l<j) qsort(a,l,j);
}
```

链表结构

链表是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。



所谓数组，是相同数据类型的元素按一定顺序排列的集合。

区别：数组申请连续的内存空间，链表申请动态的空间。
数组的缺点：需要预先设定长度。链表可以动态申请内存。

结构体

结构体(struct)： 自定义的数据类型。

```
struct Pos
{
        double x,y;
};//定义一个包含两种变量的结构体，定义了一种新的数据类型
Pos a; //声明了类型为Pos的变量a。
a.x=0.3;
a.y=5.6;//对结构体变量a的成员变量赋值。
```

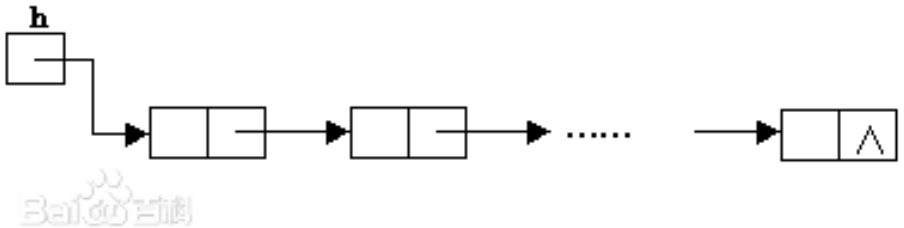
链表的节点结构

```
struct Link
{
        int data;
        Link* next;
};//定义一个包含两种变量的结构体，一个int和一个Link指针
Link *l1; //声明了类型为Link的指针变量l1。
l1 = new Link();
(*l1).data = 15; //另一种等效的写法l1->data = 15;
li->next = NULL;//NULL是空的意思，就是什么都没有。
```

链表的真面目

```
Link* head = new Link();
Link* push_back(Link* p,int x)
{
        Link* q = new Link();
        q->data = x;
        q->next = NULL;
        p->next = q;
        return p->next;
}
Link *tmp = head;
For(int i=1;i<=10;i++)
{
        tmp = push_back(tmp,i);
}
```

```
struct Link
{
        int data;
        Link* next;
};//链表节点结构
```

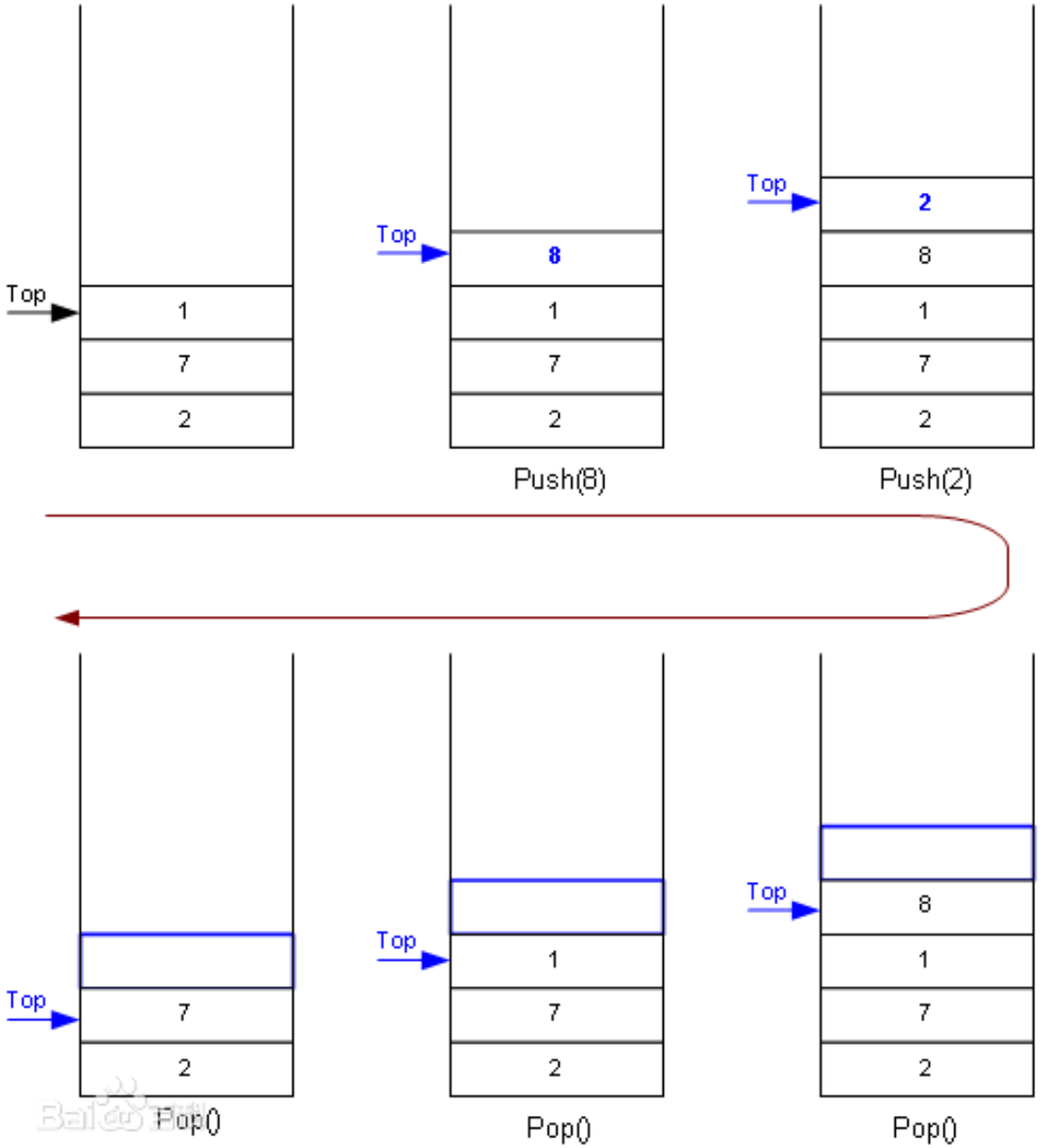


栈结构

- 栈是一种数据项按序排列的数据结构，只能在一端(称为栈顶(top))对数据项进行插入和删除。
- 特点：先进后出(FILO—First-In/Last-Out)。
- 就像羽毛球桶一样的结构，先放入的羽毛球最后取出来。
- 栈顶：top
- 栈存储结构：数组或者链表
- 两种操作：入栈 push() 和 出栈 pop()

栈

- 栈(stack)
- 栈顶： top
- 入栈 push()
- 出栈 pop()



数组栈

- 栈(stack)
- 栈顶：top
- 入栈 push()
- 出栈 pop()

```
struct stack
{
    int tot = 0;
    int s[N];
    int top()
    {
        return s[tot];
    }
    void push(int x)
    {
        s[++tot] = x;
    }
    void pop()
    {
        tot--;
    }
    bool empty()
    {
        return tot == 0;
    }
};
```

链表栈

- 栈(stack)
- 栈顶：top
- 入栈 push()
- 出栈 pop()

```
struct stack
{
    Link* Top = NULL;
    int top()
    {
        return Top->data;
    }
    void push(int x)
    {
        Link* p = new Link();
        p->data = x;
        p->last = Top;
        Top = p;
    }
    void pop()
    {
        Top = Top->last;
    }
    bool empty()
    {
        return Top == NULL;
    }
```

翻转字符串

问题描述：给定字符串，输出翻转以后的字符串。
要求：利用栈结构。

字符串头文件： #include<string.c>
字符串定义： char s[100];
字符串读入： scanf(“%s”,s);
字符串输出： printf(“%s\n”,s);
字符串长度： int len = strlen(s);

标准输入输出
Input
abcde
Output
edcba

括号合法问题

问题描述：给出一段括号串，判断是否合法，
如（（））（）是合法的，而（））（则不合法。

请思考如果上题中出现了()[]{}三种括号，包括嵌套（例：形如[{()}]也是合法的），又该如何处理？

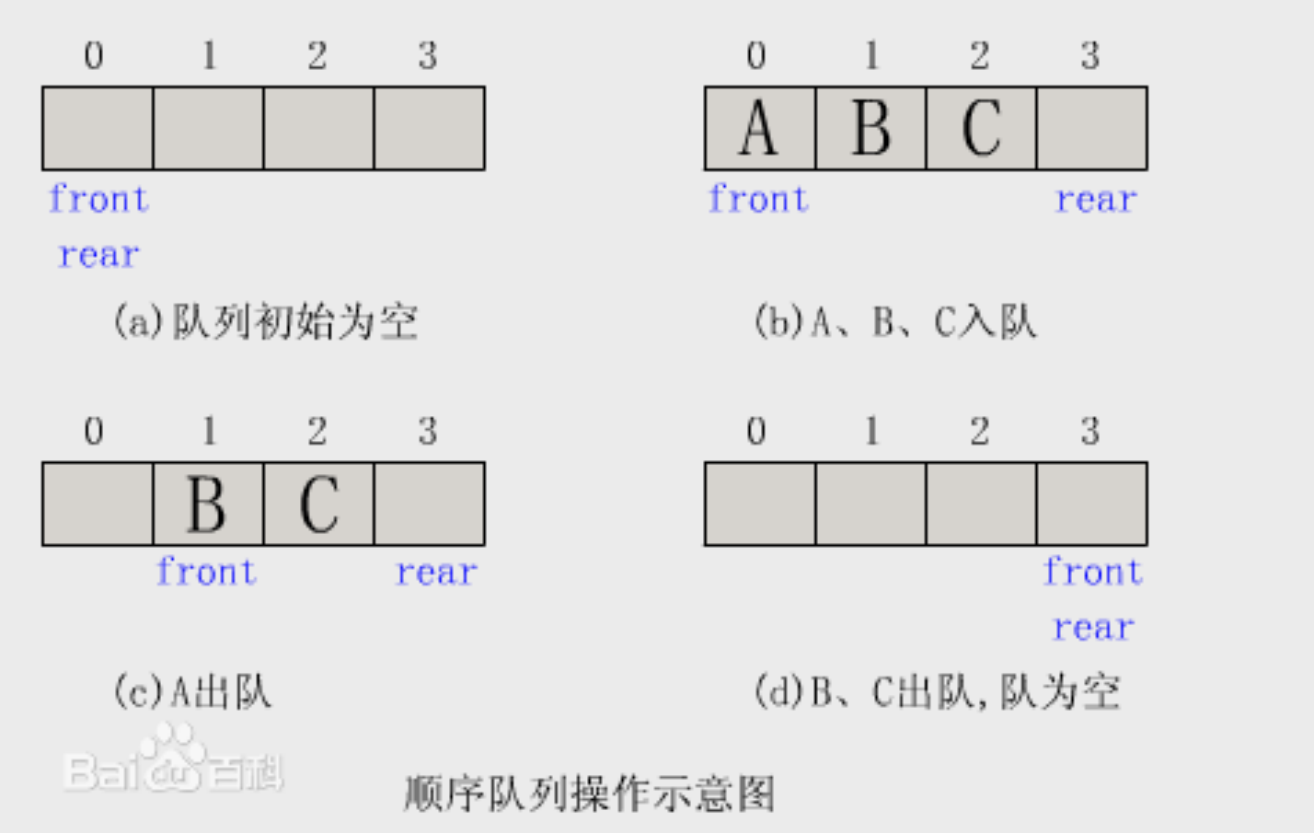
队列结构

- 队列是一种特殊的线性表，特殊之处在于它只允许在表的前端（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。
- 特点：先进先出（FIFO—first in first out）。
- 就像排队买饭。
- 队首和队尾：front 和 rear
- 队列存储结构：数组或者链表
- 两种操作：入队列 push() 和 出队列 pop()

数组队列

```
struct queue
{
    int data[N];
    int front = 0,rear = 0;
    int Front()
    {
        return data[front];
    }
    void push(int x)
    {
        data[rear++] = x;
    }
    void pop()
    {
        front++;
    }
    bool empty()
    {
        return front == rear;
    }
};
```

队列结构

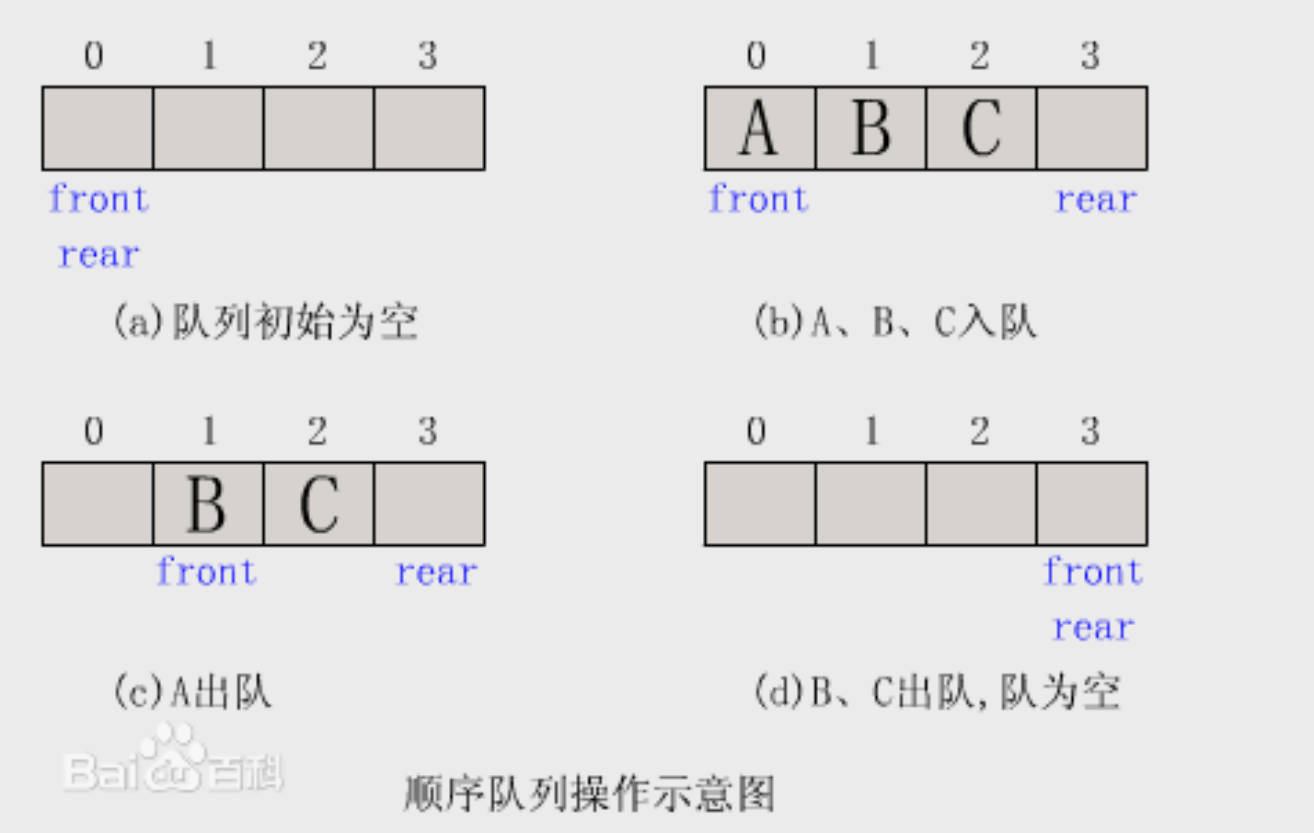


顺序队列操作示意图

链表队列

```
struct queue
{
    Link* front = &tmp,*rear = &tmp;
    int Front()
    {
        return front->data;
    }
    void push(int x)
    {
        rear->data = x;
        rear->next = new Link();
        rear = rear->next;
    }
    void pop()
    {
        front = front->next;
    }
    bool empty()
    {
        return front == rear;
    }
};
```

队列结构



STL标准模板库的使用

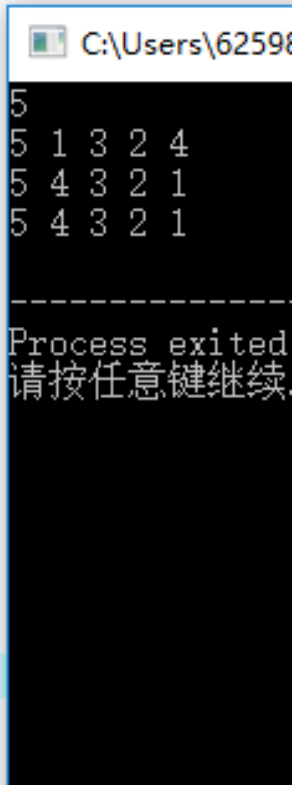
首先是栈和队列，已经被很好的实现啦，直接使用即可。

```
#include<stdio.h>
#include<stack>
using namespace std;
int main()
{
    stack<int> sta;
    for(int i=0;i<10;i++) sta.push(i);
    while(!sta.empty())
    {
        printf("%d\n",sta.top());
        sta.pop();
    }
    return 0;
}
```

```
#include<stdio.h>
#include<queue>
using namespace std;
int main()
{
    queue<int>qu;
    for(int i=0;i<10;i++) qu.push(i);
    while(!qu.empty())
    {
        printf("%d\n",qu.front());
        qu.pop();
    }
    return 0;
}
```

Sort和优先队列

```
int main()
{
    int n;
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        qu.push(a[i]);
    }
    while(!qu.empty() )
    {
        printf("%d ",qu.top());
        qu.pop();
    }
    puts("");
    sort(a,a+n,cmp);
    for(int i=0;i<n;i++) printf("%d ",a[i]);
    puts("");
    return 0;
}
```



回顾

今天你学会了什么？

一、指针和链表
二、堆栈和队列
三、STL
四、实战演练

共同进步

- 一分耕耘，一分收获！
- 希望在今后的学习生活中共同进步！

谢谢！