

## 作业三

19200444 薛辰玺

1. 给出一个算法，求循环链表中结点的个数。

```
template<class T>
int LinkListLength(LinkList<T> L)
{
    int len = 1;
    for (p = L.head->next; p != L.head; p = p->next)
        len++;
    return len;
}
```

2. 试编写算法，将元素为整数的顺序表  $(a_1, a_2, \dots, a_n)$  重新排列为以  $a_1$  为界的两部分： $a_1$  前面的值均比  $a_1$  小， $a_1$  后面的值都比  $a_1$  大，假定各个元素值互不相等。要求时间复杂度为  $O(n)$ 。

```
template<class T>
void SeparateSeqList(SeqList<T>& S)
{
    int len = S.length;
    int pt = 0;
    for(int i=1; i<len; i++)
        if (S.data[pt] > S.data[i])
        {
            swap(S.data[pt], S.data[i]);
            pt++;
        }
}
```

3. 编号为 1、2、3、4、5 的五辆列车，顺序开进一个栈式结构的站点，问开出车站的顺序有多少种可能？请具体写出所有可能的出栈序列。

用卡特兰数

$$f(5) = \frac{C_{10}^5}{5+1} = \frac{10 \times 9 \times 8 \times 7 \times 6}{6 \times 5 \times 4 \times 3 \times 2 \times 1} = 42$$

1 2 3 4 5    1 2 3 5 4    1 2 4 3 5    1 2 4 5 3    1 2 5 4 3

1 3 2 4 5    1 3 2 5 4    1 3 4 2 5    1 3 4 5 2    1 3 5 4 2

1 4 3 2 5    1 4 3 5 2    1 4 5 3 2    1 5 4 3 2

2 1 3 4 5    2 1 3 5 4    2 1 4 3 5    2 1 4 5 3    2 1 5 4 3

2 3 1 4 5    2 3 1 5 4    2 3 4 1 5    2 3 4 5 1    2 3 5 4 1

2 4 3 1 5    2 4 3 5 1    2 4 5 3 1    2 5 4 3 1

3 2 1 4 5    3 2 1 5 4    3 2 4 1 5    3 2 4 5 1    3 2 5 4 1

3 4 2 1 5    3 4 2 5 1    3 4 5 2 1    3 5 4 2 1

4 3 2 1 5    4 3 2 5 1    4 3 5 2 1    4 5 3 2 1    5 4 3 2 1

4. 利用栈实现把十进制整数转换为二至十六之间的任一进制数输出的功能。

`void DecimalConvert(int a, int n)//a为十进制数, n为进制`

```
{
    SeqStack S;
    int num;
    while (a)
    {
        num = a % n;
        S.Push(num);
        a = a / n;
    }
    while (!S.IsEmpty())
    {
        int x;
        x = S.Pop();
        if (x < 10)
            cout << x;
        else
        {
            char c = 'A' - 10;
            c += x;
            cout << c;
        }
    }
}
```

5. 设有一维数组 `stack[StackMaxSize]`, 分配给两个栈 `S1` 和 `S2` 使用, 如何分配数组空间, 使得对任何一个栈, 当且仅当数组空间全满时才不能插入。试说明你的分配方法。并分别给出两个栈各自的入栈和出栈算法。

应当将数组的两端点作为两栈底, 而栈顶在数组中动态扩展。

以 `stack[0]`为栈底的:

```
template <class T, int StackMaxSize>
void SeqStack<T, StackMaxSize>::Push()
{
    if ((!IsEmpty(data[top + 1])) || top == StackMaxSize-1)
    {
        cerr << "上溢"; exit(1);
    }
    top++;
    data[top] = x;
}

template <class T, int StackMaxSize>
T SeqStack<T, StackMaxSize>::Pop()
{
    if (top == -1)
```

```

{
    cerr << "下溢"; exit(1);
}
x = data[top];
top--;
return x;
}

```

以 `stack[StackMaxSize-1]` 为栈底的:

```
template <class T, int StackMaxSize>
```

```
T SeqStack<T, StackMaxSize>::Pop()
```

```

{
    revtop = StackMaxSize - top - 1;
    if (revtop == StackMaxSize)
    {
        cerr << "下溢"; exit(1);
    }
    x = data[revtop];
    top--;
    revtop++;
    return x;
}

```

```
template <class T, int StackMaxSize>
```

```
void SeqStack<T, StackMaxSize>::Push()
```

```

{
    revtop = StackMaxSize - top - 1;
    if ((!IsEmpty(data[revtop - 1])) || revtop == StackMaxSize - 1)
    {
        cerr << "上溢"; exit(1);
    }
    top++;
    revtop--;
    data[revtop] = x;
}

```

6. 假设表达式中允许包含 3 种括号：圆括号、方括号和大括号。试编写一个算法，检查表达式中括号是否配对，若能够全部配对则返回 1，否则返回 0。

```
stack<char> SeqStack;
```

```
string s;
```

```
bool CheckParentheses()
```

```

{
    cin >> s;
    SeqStack.push(' ');
    int L = SeqStack.size();
    int i;
    while (L--)

```

```

{
    int len = s.length();
    for (i = 0; i < len; i++)
    {
        if (s[i] != '(' && s[i] != '[' && s[i] != '{' && s[i] != ')') && s[i] != ']' &&
s[i] != '}')

            continue;
        if (s[i] == '[' || s[i] == '(' || s[i] == '{')
            SeqStack.push(s[i]);
        else if ((s[i] == ')') && SeqStack.top() == '(') || (s[i] == ']') && SeqStack.top()
== '[') || (s[i] == '}') && SeqStack.top() == '{')
            SeqStack.pop();
        else
            SeqStack.push(s[i]);
    }
    if (SeqStack.top() == '_')
        return 1;
    else
        return 0;
    while (SeqStack.top() != '_')
        SeqStack.pop();
}
}

```