

实验四 递归、串与数组

一、实验目的

1. 掌握递归算法;
2. 掌握 BF 算法与 KMP 算法,并能指出两者的区别,理解 KMP 算法对于 BF 算法的改进之处;
3. 熟悉特殊矩阵的压缩存储,掌握稀疏矩阵的三元组存储模式与转置等操作。

二、数据结构定义

串

```
#define MAX_LEN 255
typedef unsigned char
SString[MAX_LEN+1]
or
typedef char SString[MAX_LEN+1]
or
chap * p = new char[]
```

三元组

```
#define MAX_SIZE1000
typedef struct{
int i,j;
ElemType e;
}Triple;
稀疏矩阵
typedef struct{
Triple data[MAX_SIZE];
int mu,nu,tu;
}TSMatrix;
```

三、实验内容

- 1、使用递归算法编写函数完成整数划分问题将正整数 n 表示成一系列正整数之和: $n = n_1 + n_2 + \dots + n_k$, 其中 $n_1 \geq n_2 \geq \dots \geq n_k \geq 1$, $k \geq 1$

正整数 n 的这种表示称为正整数 n 的划分, 求正整数 n 的不同划分个数。

例如正整数 6 有如下 11 种不同的划分:

6;

5+1;

4+2, 4+1+1;

3+3, 3+2+1, 3+1+1+1;

2+2+2, 2+2+1+1, 2+1+1+1+1;

1+1+1+1+1+1

2、编写函数 StrAssign(), StrCopy() , StrLenth() , StrCompare() , StrConcat() , Substring(), Replace(),完成串赋值,串复制,求串长度,串比较,串连接,求子串,子串替代等相应功能。

注:Replace() 依赖 Find_KMP()

3、使用 KMP 算法,编写函数 Find_KMP(char *S,char *P,int start)实现字符串匹配。测试数据:

3.1

```
char S[] = "abcabcabcd";
```

```
char P[] = "abcabcd";
```

3.2

```
char S[] = "abcdababcbcdabcbcabcd";
```

```
char P[] = "abcabcd";
```

3.3

```
char S[] = "cocaocoaoc";
```

```
char P[] = "coaoc";
```

要求: 1.打印出模式串 P 的 next[]模式数组;

2.完成 Find_KMP()后在 Repalce()中调用,将 P 替换成“AAA”。

注意 3.2 有 2 个地方要替换。

4. (附加题) 创建三元组实现以下稀疏矩阵的存储,并利用三元组实现稀疏矩阵的转置,比较“按需查找”方法与“按位就座”方法的区别。

0	12	9	0	0	0	0
0	0	0	0	0	0	0
3	0	0	0	0	14	0
0	0	24	0	0	0	0
0	18	0	0	0	0	0
15	0	0	7	0	0	0

建议实现流程为 1.矩阵 2.三元组 3.转置的三元组 4.转置的矩阵,将 3 或 4 打印出来。