

## 第4章 排序

建议学时：6 学时

### 一. 填空题

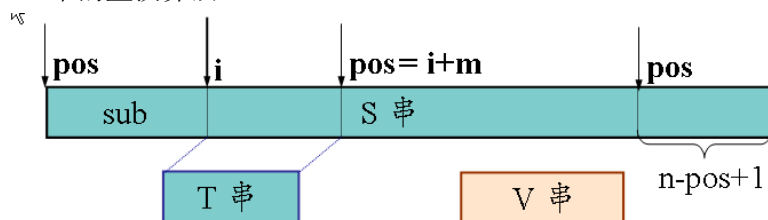
- 1、顺序存储方式和链接存储方式
- 2、两个串的长度相等且对应位置的字符相同
- 3、零个字符的串、零；由一个或多个空格字符组成的串、其包含的空格个数
- 4、14
- 5、 $LOC(A[0][0]) + (n*i+j)*k$
- 6、326
- 7、1208

### 二. 选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	B	B	D	C	D B	C	C	B	

### 三. 算法设计

- 1、串的置换算法



```
void replace(String& S, String T, String V)
{
    n=StrLength(S); m=StrLength(T); pos = 1;
    StrAssign(news, NullStr); i=1;
    while ( pos <= n-m+1 && i) {
        i=Index(S, T, pos);
        if (i!=0) {
            SubString(sub, S, pos, i-pos); // 不置换子串
```

```

        Concat(news, news, sub, V);
        pos = i+m;
    }//if
}while
    SubString(sub, S, pos, n-pos+1); // 剩余串
Concat( S, news, sub );
}
int Index (String S, String T, int pos) {
    // T 为非空串。若主串 S 中第 pos 个字符之后存在与 T 相等的子串,
    // 则返回第一个这样的子串在 S 中的 位置, 否则返回 0
    if (pos > 0) {
        n = StrLength(S);  m = StrLength(T);  i = pos;
        while ( i <= n-m+1) {
            SubString (sub, S, i, m);
            if (StrCompare(sub,T) != 0)    ++i ;
            else return i ;
        } // while
    } // if
    return 0;          // S 中不存在与 T 相等的子串
} // Index
/* StrLength(S), StrAssign(news, NullStr), SubString(sub, S, pos, i-pos), Concat(news, news,
sub, V)等算法在书中都已描述, 并且在高级语言中对字符串的操作也有相应的函数可以使用*/

```

2、 (b) ; (d)

3、

(1) GetHead[GetHead[GetTail[GetTail[(((apple)),((pear)),(banana),orange)]]]]

(2) GetHead[GetHead[GetTail[GetHead[GetTail[(apple,(pear,(banana),orange))]]]]]]