

## 第5周小结



### 串的存储结构

● 串是一种特殊的线性表,是线性表的一个子集。



- 顺序串
- 链串

### ② 链串只能采用单链表吗?

- 不一定。需要根据需要情况而定。
- 如果需要从某个节点出发前后查找,可以采用双链表。
- 如果需要快速查找尾节点,可以采用循环双链表。

# 2 串的算法设计

● 串的基本算法设计

借鉴线性表的算法设计方法。

- 顺序串 ⇔ 顺序表
- 链 串 ⇔ 单链表

### ② 串的模式匹配算法设计

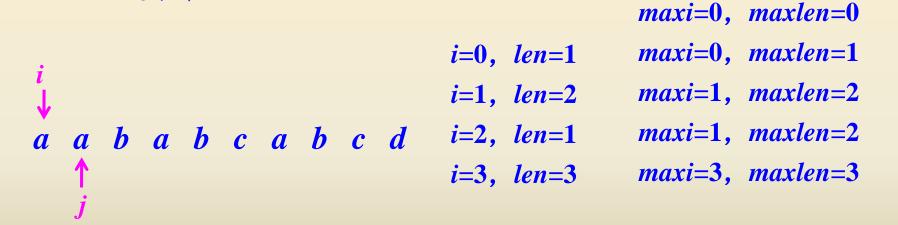
- BF算法
- KMP算法



- @ 为什么KMP算法平均性能更高?
- @ 是不是任何情况下KMP算法都好于BF算法?

《一般设串采用顺序结构存储。设计一个算法求串 8中出现的第一个最长重复子串的下标和长度。

解: (i, len) 记录当前重复子串, (maxi, maxlen) 记录第一个最长重复子串。





#### 算法如下:

```
void maxsubstr(SqString s, SqString &t)
{ int maxi=0, maxlen=0, len, i, j, k; i=0; while (i<s.length) //从下标为i的字符开始
{ j=i+1; //从i的下一个位置开始找重复子串
```

```
while (j<s.length)
   if (s.data[i]==s.data[j]) //找一个子串, 其起始下标为i, 长度为len
       len=1:
       for (k=1;s.data[i+k]==s.data[j+k];k++)
           len++;
       if (len>maxlen) //将较大长度者赋给maxi与maxlen
           maxi=i;
           maxlen=len;
       j+=len;
    else j++;
                         //继续扫描第i字符之后的字符
i++;
```

```
int main()
     SqString s,t;
     StrAssign(s,"aababcabcd");
    printf("s:"); DispStr(s);
     maxsubstr(s,t);
     printf("t:");DispStr(t);
     return 1;
```

