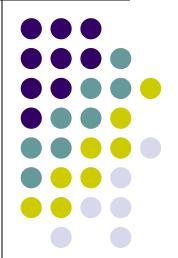
数据结构第7章习题





• 设文件(R1, R2, ..., Rn)以单链表方式表示, 指针变量FIRST指向表头结点,且表中的结点 结构为:

其中KEY为该结点的关键词域,LINK为链接域。请给出这种线性表的直接插入排序算法,并要求算法的时间复杂度为O(n²),且算法是稳定的。

```
算法InsertSort(first. first) // first指向表头哨位结点
IS1[边界]
   IF link(first)=NULL THEN RETRUN.
   IF link(link(first))=NULL THEN RETRUN.
IS2[插入排序]
   q1 \leftarrow first. \ q \leftarrow link(q1).
   WHILE q \neq NULL DO (
      if key(q1) \le key(q) then (
         q1 \leftarrow q. q \leftarrow link(q1).
      p1 \leftarrow first. p \leftarrow link(first).
      WHILE key(p) < key(q) DO (
         p1 \leftarrow p. p \leftarrow link(p).
      link(q1) \leftarrow link(q). link(p1) \leftarrow q.
      link(q) \leftarrow p. \ q \leftarrow link(q1).
```



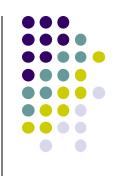
• 讨论冒泡排序算法的稳定性。



冒泡排序中,每一趟冒泡,相邻的关键词只有满足Rj>Rj+1时才会发生交换,关键词相同的记录不会发生交换,即关键词相同的元素的相对位置不变。因此冒泡排序算法是稳定的。



• 奇偶交换排序算法的基本思想描述如下: 排序 过程通过对文件x[i](1≤i≤n)的若干次扫描来完 成, 第奇数次扫描, 对所有下标为奇数的记录 x[i]与其后面的记录x[i+1](1≤ i≤ n-1)相比较, 若x[i].KEY>x[i+1].KEY,则交换x[i]和x[i+1]的 内容: 第偶数次扫描, 对所有下标为偶数的记 录x[i]与其后面的记录x[i+1](2≤i≤n-1)相比较, 若x[i].KEY>x[i+1]. KEY,则交换x[i]和x[i+1] 之内容, 重复上述过程直到排序完成为止。



• (1)排序的终止条件是什么?

(2)完成该算法的具体设计.

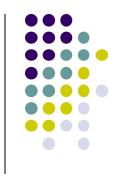
(3)分析该算法的平均运行时间.

- (1) 一趟奇偶比较无交换发生
- (2) 算法如下

```
template <typename T> void ParitySort(T X[], int |r
  for (bool change = true; change; ) {
     change = false;
     for ( int i = 0; i <= n-1; i+=2 ) { //奇交换
        if (X[i].key>X[i+1].key) {
          Swap(X[i], X[i+1]); change = true;
     for ( int i = 1; i <= n-1; i+=2 ) { //偶交换
       if (X[i].key>X[i+1].key ) {
          Swap(X[i], X[i+1]); change = true;}
```



• (3) 一次交换只能减少一个反序对,根据书上 P201反序对的平均数,平均时间为O(n²)



- 填充如下排序算法中的方框,并证明该算法能 正确性地分划.
- 算法PartA(R,s,e)

/*分划文件($R_s,R_{s+1},...,R_e$), 且 $K_{s-1}=-\infty,K_{e+1}=+\infty$ */

PA1[初始化]

$$i \leftarrow s. \ j \leftarrow \underline{1}.$$
 $e+1$
 $K \leftarrow K_s. \ R \leftarrow R_s.$

• PA2[分划过程]

WHILE i<j DO

WHILE 2 DO $j \leftarrow j-1$.

IF $i \ge j$ THEN $j \leftarrow i$.

ELSE $(R_i \leftarrow R_i$.

$$i=i+1$$
.

WHILE $K_i < K DO i \leftarrow i+1$.

IF
$$3$$
 THEN $R_i \leftarrow R_i$.)).

• PA3 <u>4</u>



$$K_j \ge K$$

i<j

$$R_j \leftarrow R$$

- ① e+1
- \bigcirc $K_j \ge K$
- 3 i<j





70 73	69	23	93	18	11	68
-------	----	----	----	----	----	----



70 73 69 23 93 18 11 68



68 | 69 | 23 | 11 | 70 | 93 | 73 18 |



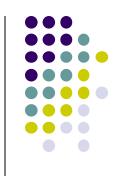
70 73 69	23 93	18 11	68
----------	-------	-------	----



70 73 69 23 93 18 11 68



11 69 23 18 70 93 73 68



• 证明: 用淘汰赛找n个元素的最大元素正好需要n—1次元素比较。



 证明:在淘汰赛中,每进行一场比赛,即进行 依次比较,都恰淘汰1个元素,找到最大元素 需要淘汰n-1个元素,因此需要n-1比较。



• 设文件(R1, R2, ..., Rn)是一个堆, Rn+1是任意一个结点。请设计一个算法, 该算法把 Rn+1添加到堆(R1, R2, ..., Rn)中, 并使(R1, R2, ..., Rn, Rn+1)成为一个堆(要求算法的时间复杂度为O(log₂n)).

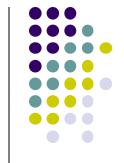
分析: 堆有大根堆和小根堆。教材上用的是大根堆。



```
算法Insert(R, n, x. R, n)
/*在堆中插入元素x,从下往上调整堆*/
|11[增加x元素]
  n \leftarrow n+1. R[n] \leftarrow x.
12[从下往上调整, 称上浮]
  i ← n.
  WHILE i>1 AND R[i/2]< R[i] DO (
     R[i/2] \leftrightarrow R[i].
     i \leftarrow i/2.
```



文件(R1, R2, ..., Rn)是一个堆, 1≤i≤n, 请给出一个算法,该算法从(R1, R2, ..., Rn)中删除 Ri,并使删除后的文件仍然是堆,要求算法的时间复杂度为O(log₂n).



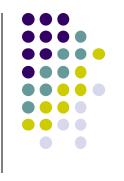
```
算法Delete(R, n, i. R, n) // 在堆中删除元素R[i], 从上
  往下调整堆
D1 [删除第i个元素]
   R[i] \leftarrow R[n]. n \leftarrow n-1.
D2 [从上往下调整, 称下沉]
  WHILE 2i + 1 < n DO (
     j ← 2i.
     IF K[j] < K[j+1] THEN j \leftarrow j+1.
     IF K[i] < K[j] THEN ( R[i]\leftrightarrowR[j]. i \leftarrow j. )
     ELSE i \leftarrow n.
  IF 2i < n AND K[i] < K[n] THEN R[i] \leftrightarrow R[n].
```

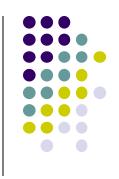


• 填充如下排序算法中的方框,并讨论该算法的稳定性.

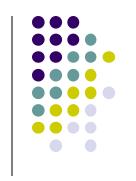
```
算法C(R, n)
/*比较计数,本算法按关键词K<sub>1</sub>、K<sub>2</sub>、...、K<sub>n</sub>
 序记录R_1、R_2、…、R_n. 一维数组COUNT[1]:
 n]用来记录各个记录的排序位置*/
[C1]
  FOR i=1 TO n DO (1) . count[i] ←1
[C2]
  FOR i=n TO 2 2 DO STEP-1
    FOR j=i-1 TO 1 STEP -1 DO
      IF 3 THEN Ki>Ki
        COUNT[j] ←COUNT[j]+1
                    count[i]← count[i]+1
```

- count[i] ←1
- STEP -1
- $K_j > K_i$
- count[i]← count[i]+1
- 算法具有稳定性





- 有一种简单的排序算法,叫做计数排序(Count Sorting). 这种排序算法对一个待排序的表(用数组表示)进行排序,并将排序结果存放到另一个新的表中。必须注意的是,表中所有待排序的关键词互不相同,计数排序算法针对表中的每个记录,扫描待排序的表一趟,统计表中有多少个记录的关键词比该记录的关键词小,假设针对某一个记录,统计出的计数值为c,那么,这个记录在新的有序表中的合适的存放位置即为c.
 - (1) 给出适用于计数排序的数据表定义。
 - (2) 对于有n个记录的表,关键词比较次数是多少?
 - (3)与简单选择排序相比较,这种方法是否更好?为 什么?



- (1) 数据表放在数组R中,元素个数为n,下标从0开始(因为小于关键词的个数为c,就放在c的位置)。
- (2) 对于每个记录,扫描待排序的表一趟,即 比较n次。一共n个记录,共需比较n²次。
- (3) 时间效率:与简单选择排序相比,比较次数增加,移动次数减少。

空间效率:辅助空间要高于简单选择排序。



THE END