

第十、十一章作业

2022211363 谢牧航

选择题

1. C
2. D
3. A
4. C
5. C
6. A
7. D
8. C
9. D
10. D
11. B
12. D
13. B

简答题

1. (503, 087, 512, 061, 908, 170, 897, 275, 653, 426)

(1) 直接插入排序

第1步: (087, 503, 512, 061, 908, 170, 897, 275, 653, 426)

第2步: (087, 503, 512, 061, 908, 170, 897, 275, 653, 426)

第3步: (061, 087, 503, 512, 908, 170, 897, 275, 653, 426)

第4步: (061, 087, 503, 512, 908, 170, 897, 275, 653, 426)

第5步: (061, 087, 170, 503, 512, 908, 897, 275, 653, 426)

第6步: (061, 087, 170, 503, 512, 897, 908, 275, 653, 426)

第7步: (061, 087, 170, 275, 503, 512, 897, 908, 653, 426)

第8步: (061, 087, 170, 275, 503, 512, 653, 897, 908, 426)

第9步: (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

(2) 希尔排序($d[1]=5$, $d[2]=3$, $d[3]=1$)

第1步 ($d=5$) : (170, 087, 512, 061, 908, 503, 897, 275, 653, 426)

第2步 (d=3) : (170, 061, 512, 087, 908, 503, 426, 275, 653, 897)

第3步 (d=1) : (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

(3) 快速排序

第1步 (基准503) : (087, 061, 170, 275, 426, 503, 908, 897, 653, 512)

第2步 (基准087) : (061, 087, 170, 275, 426, 503, 908, 897, 653, 512)

第3步 (基准512) : (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

(4) 堆排序 (用小根堆)

初始堆: (061, 087, 170, 275, 426, 512, 897, 503, 653, 908)

第1步: (087, 275, 170, 503, 426, 512, 897, 908, 653, 061)

第2步: (170, 275, 426, 503, 908, 512, 897, 653, 087, 061)

第3步: (275, 503, 426, 653, 908, 512, 897, 170, 087, 061)

第4步: (426, 503, 512, 653, 908, 897, 275, 170, 087, 061)

第5步: (503, 653, 512, 897, 908, 426, 275, 170, 087, 061)

第6步: (512, 653, 897, 908, 503, 426, 275, 170, 087, 061)

第7步: (653, 908, 897, 512, 503, 426, 275, 170, 087, 061)

第8步: (897, 908, 653, 512, 503, 426, 275, 170, 087, 061)

第9步: (908, 897, 653, 512, 503, 426, 275, 170, 087, 061)

反向: (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

(5) 归并排序

初始序列: (503, 087, 512, 061, 908, 170, 897, 275, 653, 426)

分割: (503, 087, 512, 061, 908) 和 (170, 897, 275, 653, 426)

分割: (503, 087) 和 (512, 061, 908); (170, 897) 和 (275, 653, 426)

合并: (087, 503) 和 (061, 512, 908); (170, 897) 和 (275, 426, 653)

合并: (061, 087, 503, 512, 908) 和 (170, 275, 426, 653, 897)

合并: (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

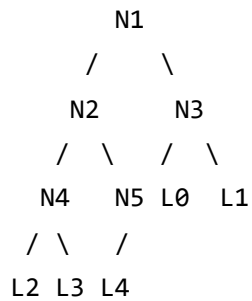
(6) 基数排序

按百位数排序: (087, 061, 170, 275, 426, 503, 512, 653, 897, 908)

按十位数排序: (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

按个位数排序: (061, 087, 170, 275, 426, 503, 512, 653, 897, 908)

2. 败者树中的败者是比较中不符合要找的值的节点，如果要找最小值就是较大的反之亦然。b是败者。败者树留下的是败者，每次调整只需和父节点比较，减少了比较次数。5路归并的败者树是一个有五个叶子的完全二叉树。败者树的叶子节点是待归并的文件，非叶子节点是败者树的内部节点。



算法题

1.

```

void selectionSort(struct ListNode *head) {
    struct ListNode *i, *j, *min;
    for(i = head; i != NULL && i->next != NULL; i = i->next) {
        min = i;
        for(j = i->next; j != NULL; j = j->next) {
            if(j->val < min->val) {
                min = j;
            }
        }
        ListNode *tmp = new ListNode();
        tmp->val = j->val;
        j->pre->next = j->next;
        tmp->next = i;
        i->pre->next = tmp;
    }
}
  
```

2.

```
void bubbleSortOptimized(int arr[], int n) {  
    int i, lastSwap;  
    int boundary = n - 1; // 初始边界为数组的最后一个元素的位置  
  
    do {  
        lastSwap = 0;  
        for (i = 0; i < boundary; i++) {  
            if (arr[i] > arr[i + 1]) {  
                // 交换元素  
                int temp = arr[i];  
                arr[i] = arr[i + 1];  
                arr[i + 1] = temp;  
                lastSwap = i;  
            }  
        }  
        boundary = lastSwap;  
    } while (boundary > 0); // 当没有交换发生时，排序完成  
}
```