

16337341_朱志儒_数据结构作业（六）

- 1、（1）从未排序序列中挑选元素，并将其依次放入到已排序序列中(初始时空)的一端的方法是 插入排序。
- （2）在待排序的元素基本有序的前提下，效率最高的排序方法是 直接插入排序。
- （3）从未排序序列中依次取出元素与已排序序列(初始时空)中的元素进行比较，将其放入已排序序列的正确位置的方法是 直接插入排序。
- （4）设有 1000 个元素，希望采用最快的速度挑选出其中前 10 个最大的元素，最好的方法是 简单选择排序。

- 2、若对关键字序列为(54, 37, 93, 25, 17, 68, 58, 41, 76)的一组记录进行快速排序时，递归调用使用的栈所能到达的最大深度是 3，共需递归调用 10，其中第二次递归调用是对 (68, 58, 93, 76) 进行排序。

- 3、在堆排序，快速排序和归并排序中，若只从存储空间考虑，应选择堆排序；若只从排序结果的稳定性考虑，应选择归并排序；若只从平均情况下排序最快考虑，应选择快速排序。

- 4、设有关键字序列为(14, 17, 53, 35, 9, 32, 68, 41, 76, 23)的一组记录，请给出用希尔排序法(增量序列是 5, 3, 1)排序时的每一趟结果。

第一趟	14	17	41	35	9	32	68	53	76	23
第二趟	14	9	32	23	17	41	35	53	76	68
第三趟	9	14	17	23	32	35	41	53	68	76

- 5、设有关键字序列为(14, 17, 53, 35, 9, 37, 68, 21, 46)的一组记录，请给出冒泡排序法排序时的每一趟结果。

第一趟	14	17	35	9	37	53	21	46	68
第二趟	14	17	9	35	37	21	46	53	68
第三趟	14	9	17	35	21	37	46	53	68
第四趟	9	14	17	21	35	37	46	53	68

- 6、设有关键字序列为(14, 17, 53, 35, 9, 37, 68, 21, 46)的一组记录，利用快速排序法进行排序时，请给出以第一个记录为基准得到的一次划分结果。

9	14	53	35	17	37	68	21	46
---	----	----	----	----	----	----	----	----

- 7、设关键字序列为(14, 17, 53, 35, 9, 37, 68, 21)的一组记录，请给出按非递增采用堆排序时的每一趟结果。

第一趟	68	21	53	17	9	35	37	14
第二趟	68	53	17	37	14	9	21	35
第三趟	68	53	37	17	35	14	9	21
第四趟	68	53	37	35	17	21	14	9
第五趟	68	53	37	35	21	14	17	9
第六趟	68	53	37	35	21	17	9	14
第七趟	68	53	37	35	21	17	14	9

- 8、设关键字序列为(314, 617, 253, 335, 19, 237, 464, 121, 46, 231, 176, 344)的一组记录，请给出采用基数排序时的每一趟结果。

第一趟	121	231	253	314	344	464	335	46	176	237	617	19
第二趟	314	617	19	121	231	335	237	344	46	253	464	176
第三趟	19	46	121	176	231	237	253	314	335	344	464	617

- 9、将哨兵放在 $R[n]$ 中，被排序的记录存放在 $R[1 \cdots n-1]$ 中，重写直接插入排序算法。

```
void InsertSort(SqList &list) {
    for (int i = list.lenth - 2; i > 0; --i) {
        if (list.key[i] > list.key[i + 1]) {
            list.key[list.lenth] = list.key[i];
            int j = i + 1;
            for (; list.key[j] < list.key[list.lenth]; ++j) {
                list.key[j - 1] = list.key[j];
            }
            list.key[j - 1] = list.key[list.lenth];
        }
    }
}
```

- 10、实际中常采用单链表存储数据记录，请写出排序记录的结构定义并修改。

结构的定义：

```
struct Node {
    int key;
    int value;
    Node *next;
    bool operator<(struct Node b) const { return key < b.key; }
};
```

修改：

```
void exchange(Node &a, Node &b) {
    int tmp = a.key;
    a.key = b.key;
    b.key = tmp;
}
```