哈尔滨工业大学（深圳）2021 年春《数据结构》

第四次作业 查找与排序

学号	姓名	成绩
200110619	梁鑫嵘	

1 简答题

1.1 第一题

对下面的关键字集{30，15，21，40，25，26，36，37}若查找表的装填因子为0.8，采用线性探测再散列方法解决冲突，完成下列内容：

1. 设计哈希函数；
2. 画出哈希表；
3. 计算查找成功和查找失败的平均查找长度。

> 1. 表的装填因子为0.8，所以表长为$\frac{8}{0.8}=10$。
>
>   因为采用线性探测再散列方法解决冲突，故设计哈希函数如下：
>
> $$hash(k)=(\lfloor\frac{k}{10}\rfloor)\ mod\ 10$$
>
>   使用散列函数计算桶号：
>
> $$H_0=hash(k)$$
>
>   一旦发生冲突，在表中顺次寻找下一个空桶桶号：
>
> $$H_i=(H_0+i)\ mod\ m, i=0,1,\ldots,m-1; m\text{为表长}$$
>
> 2. 依照设计的哈希函数画出哈希表为：
>
> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
> | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
> | | 15 | 21 | 30 | 40 | 25 | 26 | 36 | 37 | |
>
> 1. 如果查找成功，则查找的Key在表中。得到如下的查找次数表：
>
> | Key | 30 | 15 | 21 | 40 | 25 | 26 | 36 | 37 |
> | --- | --- | --- | --- | --- | --- | --- | --- | --- |
> | 次数 | 1 | 1 | 1 | 1 | 4 | 5 | 5 | 6 |
>
> 平均查找次数为$\frac{1+1+1+1+4+5+5+6}{8}=\frac{29}{8}=3.625$。
>
> 如果查找失败，则查找的Key不在表中。得到如下的查找次数表：

$[Key]$	0	1	2	3	4	5	6	7	8	9
次数	1	9	8	7	6	5	4	3	2	1

平均查找次数为：$\frac{1+9+8+7+6+5+4+3+2+1}{10}=\frac{46}{10}=4.6$。

1.2 第二题

在执行某种排序算法的过程中出现了排序码朝着最终排序序列相反的方向移动，从而认为该排序算法是不稳定的，这种说法对吗？为什么？

> 这种说法不对。
>
> 排序算法的稳定性的定义为：
>
> > 假定在待排序的记录序列中，存在多个具有相同的关键字的记录，若经过排序，这些记录的相对次序保持不变，即在原序列中，r_i=r_j，且r_i在r_j之前，而在排序后的序列中，r_i仍在r_j之前，则称这种排序算法是稳定的；否则称为不稳定的。
>
> 排序码的移动方向和算法的稳定性无关，算法稳定性和相同关键字排序前后顺序有关。所以这种说法不对。

1.3 第三题

设有5个互不相同的元素a,b,c,d,e，能否通过7次比较就将其排好序？如果能，请列出其比较过程；如果不能，则说明原因。

> 能。
>
> 对其中四个元素a,b,c,d排序。假设a与b比较，$a>b$；c与d比较，$c>d$。然后再比较a和c，得到①$c>a>b$且$c>d$，或者②$a>c>d$且$a>b$。通过三次比较确定了四个元素中三个元素的大小关系和一对另外元素的大小关系。
>
> 假设得到大小关系为②。通过折半查找将e放入$a>c>d$中，需要两次比较。假设$e>c$，有下面两种情况：①$e>a>c>d$，②$a>e>c>d$。
>
> 情况①，b需要和c、d比较，比较两次；情况②，b需要和c比较之后再和e或者d比较。
>
> 综上，5个元素总共需要比较7次。

1.4 第四题

设有6个有序表A、B、C、D、E、F，分别含有10、35、40、50、60和200个数据元素，各表中元素按升序排列。要求通过5次两两合并，将6个表最终合并成1个升序表，并在最坏情况下比较的总次数达到最小。请回答下列问题：

1. 给出完整的合并过程，并求出最坏情况下比较的总次数。
2. 根据你的合并过程，描述n（$n \geq 2$）个不等长升序表的合并策略，并说明理由。

> 1. 归并排序是稳定的排序算法，最差时间复杂度为$O(n\log n)$。对于两个有序表的合并操作，设两个表长度分别为a,b，则最坏情况下需要比较的次数为$a+b-1$，最好情况下比较次数为$min(a,b)$。
>
>   所以最坏情况下比较总次数$w=(10+35+40+50+60+200)-5=390$次。
> 2. 策略：应该尽量先选择长度相近的最短的表进行合并操作。
>
>   原因：虽然选择不同的时候在最坏情况下，合并比较次数是一样的，但是选择长度相近的表进行合并，可能的最好情况比较次数$min(a,b)$能更加小；同时，先选择短的表合并能够给长度较长的表提供长度相近的选择。

1.5 第五题

选择排序、插入排序、希尔排序、快速排序、归并排序、堆排序和基数排序，哪些排序是不稳定的，为什么，请举例说明。

> 不稳定的排序以及举例：
>
> - 快速排序
>
> $[5,4,2,2,3,1]$，如果选择第二个2作为中间比较值，则第一个2可能被放到比较大的一边，使得排序前后两个2的顺序发生改变。
> - 选择排序
>
> $[3,5,3,2,6]$，$3 \Longleftrightarrow 2$，然后选择5和3交换顺序，但是两个3都能选择，会破坏两个3的先后顺序。
> - 希尔排序
>
> $[4,2,2,1]$，选择步进为2，$\Rightarrow [2,1,4,2]$，再选择步进为$[1,2,2,4]$。可以看到两个2在排序前后交换了位置，所以希尔排序不稳定。
> - 堆排序
>
>   假设已经排好序的小根堆为$[1,3,2,3]$，此时1出队，下一个出队的可能是第一个3，也可能是第二个3，所以堆排序是不稳定的。

2 算法设计

设计一个算法，使得在尽可能少的时间内重排数组，将所有负值的关键字放在所有非负值的关键字之前。

> 此情景和快速排序中选择比较值然后把数组分作大于小于比较值的两部分一样，所以详细方法是：
>
> 1. 从数组开头和末尾开始向中间扫描
> 2. 开头开始扫描到正的关键字，从末尾开始扫描到负的关键字
> 3. 重复2，直到两头扫描到同一个位置

```
void split_array(int *data, size_t size) {
  size_t i = 0, j = size - 1;
  while (i < j) {
    while (i < j && data[i] < 0) i++;
    while (i < j && data[j] > 0) j--;
    std::swap(data[i], data[j]);
  }
}
```