Homework02 2021.10.13

1

下面的排序算法中哪些是稳定的：插入排序、归并排序、堆排序、快速排序和计数排序？给出一个能使任何排序算法都稳定的方法。你所给出的方法带来的额外时间和空间开销是多少？

插入排序、归并排序、计数排序时稳定的排序算法，而堆排序、快速排序则是不稳定的排序算法。

为序列中的每一个元素另外设置一个标记域，用来区分数值域相同的元素。例如对于序列中连续的4，在处理之后序列称为`{4,1}{4,2}{4,3}{4,4}{4,5}...{4,k}...`，在排序时先使用原来的排序算法进行排序，然后使用稳定的排序算法（如插入排序）根据标记域，对数值域相同的元素进行排序。

- 额外的时间开销：第一次排序结束遍历序列查找数值域相同的区域+使用稳定排序算法对该区域进行排序。
- 额外的空间开销：为每一个序列元素新建额外的标记域。

2

假设所有元素都是互异的，说明在最坏情况下。如何使快速排序的运行时间为 $O(nlogn)$。

快速排序的运行时间取决于基准元素的选取

若算法的每一次递归调用过程中，划分都是最大程度不平衡的，即划分产生的两个子问题分别包含了0个和$n-1$个元素，就是快速排序的最坏情况，此时算法运行时间的递归式可以表示为：
$T(n)=T(n-1)+O(n)$
最坏情况下的时间复杂度为$O(n^2)$

如果每次都选择中位数为基准元素，那么运行时间的递归式为：$T(n)=2T(n/2)+O(n)$
时间复杂度为 $O(n\lg n)$。

3

给定一个整数数组，其中不同的整数所包含的数字的位数可能不同。但该数组中，所有整数中包含的总数字位数为 n。设计算法使其可以在 $O(n)$ 时间内对该数组进行排序。

算法思想：假设数组中元素的个数为count。利用桶排序的思想，将数组中位数相同的元素归为一组，再使用基数排序对每组中的元素进行精细排序。其中第一步的时间复杂度为$O(n)$，第二步的时间复杂度为$O(n)$。总的时间复杂度为$O(n)+O(n)=O(n)$

```
SORT(A, count)
for i = 1 to count do
    k = DIGIT-NUMBER(A[i])
    B[k][C[k]] = A[i]
    C[k]++
    MAX(k, max)
for i = 1 to max do
    RADIX-SORT(B[i], C[i])

DIGIT-NUMBER(k)
i = 1
```

```
while k / pow(10, i) != 0 do
    i++
return i

RADIX-SORT(A,d)
for i = 1 to d do
    use a stable sort to sort array A on digit i
```

4

SELECT 算法最坏情况下的比较次数 $T(n)=\Theta(n)$，但是其中的常数项是非常大的。请对其进行优化，使其满足：
• 在最坏情况下的比较次数为 $\Theta(n)$。
• 当 i是小于 $n/2$ 的常数时，最坏情况下只需要进行 $n+O(logn)$ 次比较。

假设输入数组为A[p - p+n-1]共n个元素

1. 取$m=\lfloor n/2\rfloor$,将输入数组进行划分，第一组第1...m个数，第二组为第m+1...2m个数，如果n为奇数，还会多一个数未分组A[p+2m+1].
2. 分别对两部分的第j个数进行比较（$0\le j\le m$)，就是A[p]和A[p+m],A[p+1]和A[p+m+1],...,如果A[p+j]<A[p+m+j],则交换它们，即把较小元素放到A[p+m+j]，较大元素放到A[p+j].
3. 对A[p+m]...A[p+n-1]递归执行1、2步骤。如果递归中对两个分组的元素进行调换，那么他们上一层分组左侧组相应位置也得调换，依次类推，上上层以及更多层的相应位置都得调换，保证每个分组左侧一组依次大于右侧一组。
4. 每递归一次，分组内元素的数量就减少一半。当i大于等于分组元素个数的一半时，停止递归，采用SELECT方法对最后分组L划分，此时L前i个数是该分组最小的i个数，倒数第二个分组L1元素依次大于L，那么L和L1的第i小数就在L的前i个数和L1的前i个数之中，利用SELECT对这2i个数操作(如果n为奇数，就将最后一个数也加进去），找到这两分组前i小的数，返回上一层调用当中。
5. 在上一层调用中，用SELECT对左右分组前i个数共2i个数操作，然后再返回，直到顶层。