

数据结构与算法I 作业5

2019201409 于倬浩

8.2-4

将一个长度为 $k+1$ 的数组 $sum[]$ 初始化为全0。接下来遍历每个数，当遍历到第 i 个数时，令 $sum[a[i]] = sum[a[i]] + 1$ 。

遍历完所有数字后，从1遍历到 k ，令 $sum[i] = sum[i] + sum[i-1]$ 。

接下来，当我们处理查询 $[a, b]$ 时，若 a 不为0则答案即为 $sum[b] - sum[a-1]$ ，否则为 $sum[b]$ 。

8.3-5

设处理 n 个 k 进制数字，每个数字共 d 位。此算法按照高位到低位的顺序进行每轮排序，每轮排序中又递归地将每一堆卡片再分为 k 堆，接下来递归进入下一轮，最后合并。

因此，最坏情况下，每一轮排序涉及到的所有堆都非空，共需进行 d 轮排序，需要的操作次数为 $\Theta(nd)$ ，其中涉及到 $\Theta(k^d)$ 堆卡片。

8.4-2

当输入的数值集中在同一个区间时，该算法会对这个区间内的 n 个数进行插入排序，而插入排序的时间复杂度为 $\Theta(n^2)$ ，因此该算法在最坏情况下的时间复杂度为 $\Theta(n^2)$ 。

解决方法是，在处理被分出的小区间时，采用 $\Theta(n \lg n)$ 时间复杂度的原址排序，例如堆排序，则即使所有数值集中在同一个区间时，仍可保证时间复杂度不会退化到 $\Theta(n^2)$ 。