**算法读书笔记——计数排序**

计数排序是一个非基于比较的排序算法，该算法于1954年由 Harold H. Seward 提出。它的优势在于在对一定范围内的整数排序时，它的复杂度为Ο(n+k)（其中k是整数的范围），快于任何比较排序算法。当然这是一种牺牲空间换取时间的做法，而且当O(k)>O(n\*log(n))的时候其效率反而不如基于比较的排序（基于比较的排序的时间复杂度在理论上的下限是O(n\*log(n)), 如归并排序，堆排序）。

**算法思想：**

计数排序对输入的数据有附加的限制条件：

1、输入的线性表的元素属于有限偏序集S；

2、设输入的线性表的长度为n，|S|=k（表示集合S中元素的总数目为k），则k=O(n)。

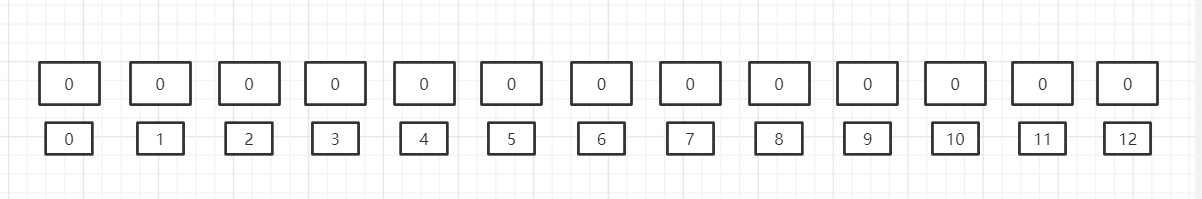
在这两个条件下，计数排序的复杂性为O(n)。

计数排序的基本思想是对于给定的输入序列中的每一个元素x，确定该序列中值小于x的元素的个数（此处并非比较各元素的大小，而是通过对元素值的计数和计数值的累加来确定）。一旦有了这个信息，就可以将x直接存放到最终的输出序列的正确位置上。例如，如果输入序列中只有17个元素的值小于x的值，则x可以直接存放在输出序列的第18个位置上。当然，如果有多个元素具有相同的值时，我们不能将这些元素放在输出序列的同一个位置上，因此，上述方案还要作适当的修改。

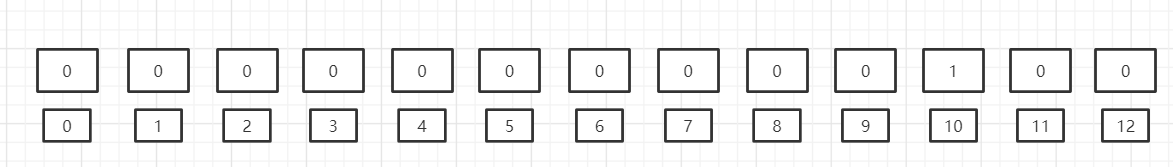
**案例设计**

有一堆整数为：10，2，3，5，6，8，7，9，5，4，6，12，6，7，6，9，11，2，4，1，0对其进行排序

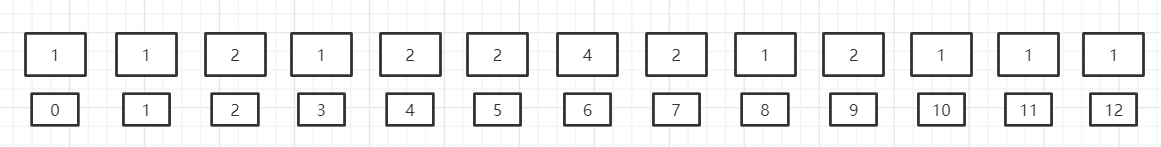
1、首先找到这一组数据的数据范围，取值范围为0到12，所以我们建立一个长为13的数组。并将所有数据的初始值设置为0.



2、从第一个值开始进行取值，将新数组和值对应的下标加上1



3、重复上一步操作直到全部值都被扫过一遍



4、再建立一个和原来数组同样长度的数组，安装上述产生的数组将数组依次输入新建数组，就完成了排序。

**算法分析：**

如果原始数列的规模是N，最大最小整数的差值是M，由于代码中第1、2、4步都涉及到遍历原始数列，运算量都是N，第3步遍历统计数列，运算量是M，所以总体运算量是3N+M，去掉系数，时间复杂度是O(N+M)。

至于空间复杂度，如果不考虑结果数组，只考虑统计数组的话，空间复杂度是O(M)。

**总结**

**计数算法存在局限性**

1.当数列最大最小值差距过大时，并不适用于计数排序

计数算法是在取值范围不大但取值数量很多时后比较好用的一种算法，如歌取值范围过大则会十分的占用空间。比如给定20个随机整数，范围在0到1亿之间，此时如果使用计数排序的话，就需要创建长度为1亿的数组，不但严重浪费了空间，而且时间复杂度也随之升高。

2.当数列元素不是整数时，并不适用于计数排序

如果数列中的元素都是小数，比如3.1415，或是0.00000001这样子，则无法创建对应的统计数组，这样显然无法进行计数排序。

正是由于这两大局限性，才使得计数排序不像快速排序、归并排序那样被人们广泛适用。