

# 位图排序

问题：

给出一个或者多个非负整数，整数的范围很大，甚至可以上亿。但整数之间互不相同，且没有依赖。能否用最少的空间和最短的时间将这些数按升序排列。

思想：位图排序

位图排序摒弃了我们常见的排序方法，而是使用二进制的方式来存储数据。根据[Java API BitSet](#)的文档可以得知，它是利用数组的下标来表示该元素是否出现，如果出现，该位设置为true，否则就是false。

从上面也得知，位图排序得知道排序元素的最大范围从而确定bitset数组的范围，然后如果有些数据没有出现，还是得为这些数据保留一个位，因此位图排序适用于高度密集出现的不相同的大范围排序，比如说电话号码的排序。

根据API文档，bitset给定参数是n的时候，表示的范围其实是[0, n-1],因此如果知道最大数，参数必须得加1.所以对{1,2,3,5,8,13} 进行位图排序，bitset长度为14，且其设置应为[0,1,1,1,0,1,0,0,1,0,0,0,0,1].这里我用1表示true，0表示false。

因此如果要对数据范围为 $10^7$ 进行排序，那么我们需要的内存空间是：

$$10^7 / (1024 * 1024 * 8) = 1.2M$$

而且运行时间是线性的。

看代码

```
1. public class BitSort {
2.     final static int MAX = 10000000;
3.     public static void bitSort(int[] A) {
4.         BitSet set = new BitSet(MAX + 1);
5.         for (int i = 0; i < A.length; i++) {
6.             set.set(A[i]);
7.         }
8.         for (int i = 0; i < MAX + 1; i++) {
9.             if (set.get(i)) {
10.                System.out.println(i);
11.            }
12.        }
13.    }
14.
15.    public static void main(String[] args) {
16.        int[] A = {5,1,2,13,7,10,0,20,16,9};
17.        bitSort(A);
18.    }
19. }
20. 在eclipse上测试通过
```