意外了解到基数排序，发现这算法太厉害了！

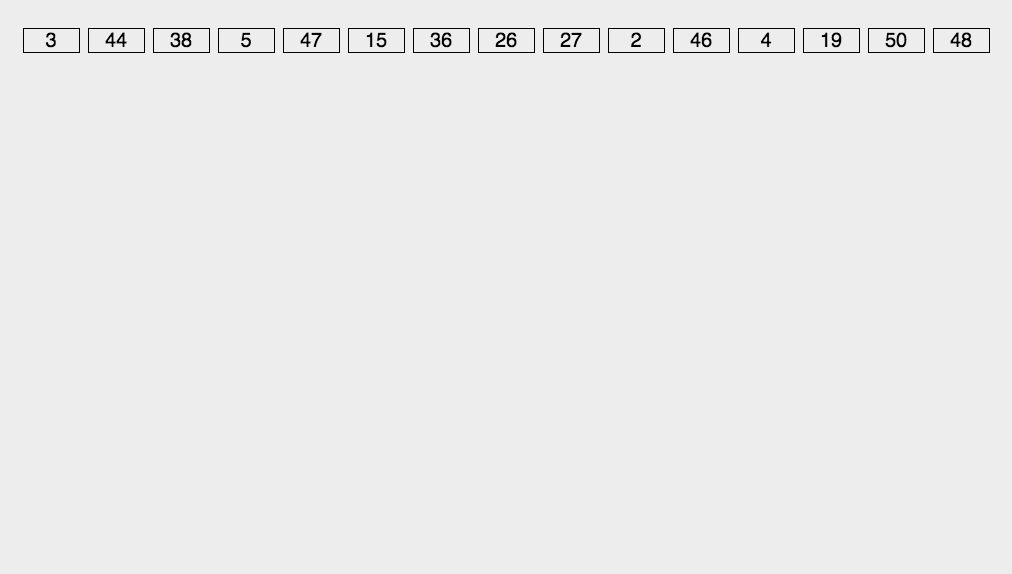
这种排序算法的时间复杂度为Θ(d(n+k))，其中n表示待排序列的规模，d表示待排序列的最大位数，k表示每一位数的范围，这也是一种时间换空间的算法。

适用情况

我们看这个算法的时间复杂度我们应该就可以猜出来，这个算法适用于位数不多，待排序列最大位数不是特别大，每一位数的范围不大的情况下（当然对于数字排序，每一位的范围都是[0,9]），这时候排序速度非常快。

基本思路

基数排序的思路是按照低位先排序，然后收集；再按照高位排序，然后再收集；依次类推，直到最高位。有时候有些属性是有优先级顺序的，先按低优先级排序，再 按高优先级排序。最后的次序就是高优先级高的在前，高优先级相同的低优先级高的在前。基数排序基于分别排序，分别收集，所以是稳定的。



至此，我们的基数排序就完成了。理解起来感觉很简单。至于为什么不是从高位到低位呢？我觉得实际上理解起来也很简单：比如我们还是对上面的序列排序，第一轮我们对最高位排序，然后对十位排序的时候就把之前对最高位排序之后的有一定顺序的序列给打乱了。因此从高位到低位排序是错误的。

那么我们如何证明这种排序的正确性呢？我们可以使用归纳法：假设待排的n个数有d位，那么n个数中的所有d-1位数已经排好序了，我们现在开始对第t位开始使用排序，有两种可能的情况，如果n个数中的t位有相同的数，那么我们应该假设之前排序是稳定的，对于相同的数，先出现的排在前面，后出现的排在后面。如果n个数中的第t位不相同，那么我们对其再次进行计数排序。直到所有d位数都排好了。

通过我们的证明也可以看出，我们对基数排序中的每一位的排序算法必须是稳定的。

原文链接：https://blog.csdn.net/green\_hand2333/article/details/106648879