[算法设计-全排列递归](https://www.cnblogs.com/xmfdsh/p/4048086.html)

排列：从n个元素中任取m个元素，并按照一定的顺序进行排列，称为排列；

全排列：当n==m时，称为全排列；

比如：集合{ 1,2,3}的全排列为：

{ 1 2 3}

{ 1 3 2 }

{ 2 1 3 }

{ 2 3 1 }

{ 3 2 1 }

{ 3 1 2 }

我们可以将这个排列问题画成图形表示，即排列枚举树，比如下图为{1,2,3}的排列枚举树，此树和我们这里介绍的算法完全一致；

算法思路：

(1)n个元素的全排列=（n-1个元素的全排列）+（另一个元素作为前缀）；

(2)出口：如果只有一个元素的全排列，则说明已经排完，则输出数组；

(3)不断将每个元素放作第一个元素，然后将这个元素作为前缀，并将其余元素继续全排列，等到出口，出口出去后还需要还原数组；

**有重复元素的排列问题**

然后现在的题目要求是排列中的元素是包含相同元素的，给定n以及待排的n个可能重复的元素。计算输出n个元素的所有不同排列，因此上面那个算法显然还是不够好，因为相同的元素都当成不同的元素，因此有了重复的排列在里面

去掉重复符号的全排列：在交换之前可以先判断两个符号是否相同，不相同才交换，这个时候需要一个判断符号是否相同的函数。也就是下面的IsSwap（）；

对122，第一个数1与第二个数2交换得到212，然后考虑第一个数1与第三个数2交换，此时由于第三个数等于第二个数，所以第一个数不再与第三个数交换。再考虑212，它的第二个数与第三个数交换可以得到解决221。

去掉重复的规则：**去重的全排列就是从第一个数字起每个数分别与它后面非重复出现的数字交换。**

**非递归的实现**

实现思路：

要考虑全排列的非递归实现，先来考虑如何计算字符串的下一个排列。如"1234"的下一个排列就是"1243"。只要对字符串反复求出下一个排列，全排列的也就迎刃而解了。

如何计算字符串的下一个排列了？来考虑"926520"这个字符串，我们从后向前找第一双相邻的递增数字，"20"、"52"都是非递增的，"26 "即满足要求，称前一个数字2为替换数，替换数的下标称为替换点，再从后面找一个比替换数大的最小数（这个数必然存在），0、2都不行，5可以，将5和2交换得到"956220"，然后再将替换点后的字符串"6220"颠倒即得到"950226"。

**总结：**

排列在笔试面试中很热门，在百度和迅雷的校园招聘以及程序员和软件设计师的考试中都考到了，因此了解全排列算法对我们都很有好处。也是算法的一个基本思想。递归算法的思路比较直，而非递归的就比较难去想到使用这种方法来实现。

1．**全排列就是从第一个数字起每个数分别与它后面的数字交换。**

2．**去重的全排列就是从第一个数字起每个数分别与它后面非重复出现的数字交换。**

3．**全排列的非递归就是由后向前找替换数和替换点，然后由后向前找第一个比替换数大的数与替换数交换，最后颠倒替换点后的所有数据。**