**下一个排列**

**字典序计算下一个排列**

利用字典序计算全排列中下一个排列的具体方法如下：

设是的一个全排列：

1）从排列的右端开始，找出第一个比右边数字小的数字的序号（从左端开始计算），即 

2）在的右边的数字中，找出所有比大的数中最小的数字，即（右边的数从右至左是递增的，因此是所有大于的数字中序号最大者）

3）对换，

4）再将倒转得到排列，这就是排列的下一个排列。

**下一个排列样例**

给出排列[1,3,2,3]，其下一个排列是[1,3,3,2]

给出排列[4,3,2,1]，其下一个排列是[1,2,3,4]

Lintcode测试链接：

排列中可能包含重复的整数：<https://www.lintcode.com/problem/next-permutation/description>

<https://www.lintcode.com/problem/next-permutation-ii/description>

已通过的代码见nextPermutation.cpp

**上一个排列**

**字典序计算上一个排列**

具体方法与下一个排列相似，如下：

设是的一个全排列：

1）从排列的右端开始，找出第一个比右边数字大的数字的序号（从左端开始计算），即 

2）在的右边的数字中，找出所有比小的数中最大的数字，即（右边的数从右至左是递减的，因此是所有小于的数字中序号最大者）

3）对换，

4）再将倒转得到排列，这就是排列的下一个排列。

**上一个排列样例**

给出排列[1,3,2,3]，其上一个排列是[1,2,3,3]

给出排列[1,2,3,4]，其上一个排列是[4,3,2,1]

Lintcode测试链接：

排列中可能包含重复的整数：

<https://www.lintcode.com/problem/previous-permutation/description>

已通过的代码见previousPermuation.cpp

**排列序号II（有重复的数字）**

**康托展开**



其中，为整数，表示在排列数中，在之后（也就是右边）且数值小于的个数；为康托展开值，表示小于排列数的组合有，排列数排在第位。

**康托展开样例**

在（1，2，3，4，5）5个数的排列组合中，计算 34152的康托展开值。

首位是3，则小于3的数有两个，为1和2，a[4]=2，则首位小于3的所有排列组合为 a[4]\*(5-1)!

第二位是4，则小于4的数有两个，为1和2，注意这里3并不能算，因为3已经在第一位，所以其实计算的是在第二位之后小于4的个数。因此a[3]=2

第三位是1，则在其之后小于1的数有0个，所以a[2]=0

第四位是5，则在其之后小于5的数有1个，为2，所以a[1]=1

最后一位就不用计算啦，因为在它之后已经没有数了，所以a[0]固定为0

根据公式：

X = 2 \* 4! + 2 \* 3! + 0 \* 2! + 1 \* 1! + 0 \* 0!

= 2 \* 24 + 2 \* 6 + 1

= 61

所以比 34152 小的组合有61个，即34152是排第62。

**特例：有重复的数字**

对于排列数中有重复数的时候，需要将除以重复元素个数的阶乘（相当于除以重复元素的全排列）

Lintcode测试链接：<https://www.lintcode.com/problem/permutation-index-ii/description>

已通过的代码见permutationIndexII.cpp

**第k个排列**

**逆康托展开**

对于上述例子，在（1，2，3，4，5）给出61可以算出起排列组合为 34152。由上述的计算过程可以容易的逆推回来，具体过程如下：

用 61 / 4! = 2余13，说明a[4]=2,说明比首位小的数有2个，所以首位为3。

用 13 / 3! = 2余1，说明a[3]=2，说明在第二位之后小于第二位的数有2个，所以第二位为4。

用 1 / 2! = 0余1，说明a[2]=0，说明在第三位之后没有小于第三位的数，所以第三位为1。

用 1 / 1! = 1余0，说明a[1]=1，说明在第二位之后小于第四位的数有1个，所以第四位为5。

最后一位自然就是剩下的数2啦。

通过以上分析，所求排列组合为 34152。

Lintcode测试链接：

<https://www.lintcode.com/problem/permutation-sequence/description>

已通过的代码见getPermutation.cpp