

全排列生成算法：next_permutation PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022__E5_85_A8_E6_8E_92_E5_88_97_E7_c97_646765.htm 导读:全排列的生成算法有很多种，有递归遍例，也有循环移位法等等。概念 C /STL中定义的next_permutation和prev_permutation函数则是非常灵活且高效的一种方法，它被广泛的应用于为指定序列生成不同的排列。本文将详细的介绍prev_permutation函数的内部算法。按照STL文档的描述，next_permutation函数将按字母表顺序生成给定序列的下一个较大的排列，直到整个序列为降序为止。prev_permutation函数与之相反，是生成给定序列的上一个较小的排列。二者原理相同，仅遍例顺序相反，这里仅以next_permutation为例介绍算法。先对序列大小的比较做出定义：两个长度相同的序列，从两者的第一个元素开始向后寻找，直到出现一个不同元素（也可能就是第它们的第一个元素），该元素较大的序列为大，反之序列为小；若一直到最后一个元素都相同，那么两个序列相等。设当前序列为 p_n ，下一个较大的序列为 p_{n+1} ，这里蕴藏的含义是再也找不到另外的序列 p_m ，使得 $p_n < p_m$ 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com