山东大学 计算机科学与技术 学院

数据结构与算法 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201900150221 | 姓名： 张进华 | | 班级： 智能19 |
| 实验题目：递归练习 | | | |
| 实验学时：2 | | 实验日期： 9.14 | |
| 实验目的：   1. 熟悉开发工具的使用。 2. 掌握递归的实现思想。 | | | |
| 软件开发工具：  visual studio code  g++ | | | |
| 1. 实验内容   现有一个有 n 个元素的序列 a=[a1,a2,⋯,an]，定义其价值为∑ 𝑎𝑖 ⊕ 𝑖 𝑛 𝑖=1 给出这样一个序列，求其所有排列的价值 vi 或𝑉1|𝑉2|𝑉3|...|𝑉n-1 |𝑉𝑛 其中∣为位运算或操作，⊕为位运算异或操作。。   1. 数据结构与算法描述 （整体思路描述，所需要的数据结构与算法）   全排列可以看做固定前 i 位，对第 i+1位之后的再进行全排列，比如固定第 一位，后面跟着 n-1位的全排列。那么解决 n-1位元素的全排列就能解决 n 位元素的全排列了   1. 测试结果（测试输入，测试输出）      1. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）   这道题用递归实现了集合的全排列，交换后的序列除去第一个元 素放入到下一次递归中去了，递归完成了再进行下一次循环。这 是某一次循环程序所做的工作   1. 附录：实现源代码（本实验的全部源程序代码，程序风格清晰易理解，有充分的注释）   #include <bits/stdc++.h>  using namespace std;  int sumValue = 0;  template <class T>  void perm(T arr[], int start, int size, int &sumValue) // 计算每种排列的价值,size标志全排列个数，start标志当前选择位置  {      if (start == size)      { //全部遍历完成          int per\_value = 0;          for (int i = 0; i < size; i++)              per\_value += (arr[i] ^ (i + 1));          sumValue |= per\_value;      }      else      {          for (int i = start; i < size; i++)          {              swap(arr[start], arr[i]);              perm(arr, start + 1, size, sumValue);              swap(arr[i], arr[start]);          }      }  }  int main()  {      int n;      cin >> n;      int \*arr = new int[n];      for (int i = 0; i < n; i++) // 初始化          cin >> arr[i];      perm(arr, 0, n, sumValue);      cout << sumValue << endl;  } | | | |