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：201705130113 | 姓名： 黄瑞哲 | | 班级： 计科17.3 |
| 实验题目：递归练习 | | | |
| 实验学时：2 | | 实验日期： 2018.9.27 | |
| 实验目的：   1. 熟悉开发工具的应用 2. 掌握递归的实现思想 | | | |
| 软件环境：  Ubuntu 18.04  G++ 7.3.0  Vim 8.0 | | | |
| 1. 实验内容（题目内容，输入要求，输出要求） 2. 键盘输入n和n个互不相同的整数，输出n个数的全排列 3. 键盘输入n和n个互不相同的整数，输出n个整数的所有子集 4. 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）   （1）  利用了递归的思想。枚举第i位是哪一个元素，然后递归处理剩余的序列。当前元素可以是之后的某一个元素，所以可以利用swap把现在的元素和之后的某个元素进行交换，等到函数回溯是在交换回来以供之后再次进行交换。  （2）  利用了递归的思想。递归的枚举第i个元素是否出现在当前这个子集中。枚举的过程中使用了二进制来记录当前元素是否出现，如果flag二进制第i位上是1表示选取第i个元素，如果为0则表示不选取第i个元素，当k==n的时候枚举结束，根据flag的二进制表示来输出最终的子集。   1. 测试结果（测试输入，测试输出，结果分析）   （1）    （2）     1. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径） 2. 程序会输出n！行数据，每一行输出的数字表示该排列每个数字出现的情况。不存在问题 3. 程序会输出2^n行数据，第一行数据为空表示空集，最后一行数据数量为n表示全集。未出现问题。 4. 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）   （1）  #include <iostream>  /\*\*  \* get the permutation of arr[k, n)  \*/  template<typename T>  void permutations(int k, int n, T\* arr)  {  if (k == n)  {  for (int i = 0; i < n; ++i) std::cout << arr[i] << ' ';  std::cout << std::endl;  return;  }  for (int i = k; i < n; ++i)  {  std::swap(arr[i], arr[k]);  permutations(k + 1, n, arr);  std::swap(arr[i], arr[k]);  }  }  int main()  {  int n;  std::cin >> n;  int \*num = new int[n];  for (int i = 0; i < n; ++i) std::cin >> num[i];  permutations(0, n, num);  delete []num;  return 0;  }  （2）  #include <iostream>  template<typename T>  /\*\*  \* get all subsets of arr  \* @param flag 用二进制压位来表示第i个元素是否在子集中  \* @param k arr的左端点  \* @param n arr的右端点  \* @param arr 全集  \*/  void subset(long long flag, int k, int n, const T \*arr)  {  if(k==n)  {  for(int i=0;i<n;++i)  // if in position i, the binary expression of flag is 1, then output arr[i]  if(flag>>i&1) std::cout << arr[i] << ' ';  std::cout << std::endl;  return;  }  /\*there may be 1 or 0 on this position k\*/  subset(flag, k+1, n, arr);  subset(flag|(1ll<<k), k+1, n, arr);  }  int main()  {  int n;  std::cin>>n;  int \*num = new int[n];  for(int i=0;i<n;++i) std::cin>>num[i];  subset(0, 0, n, num);  delete []num;  return 0;  } | | | |