

数据结构与算法 课程实验报告

学号：202200130048	姓名：陈静雯	班级：6
实验题目：递归练习		
实验学时：2	实验日期：2023.9.14	
实验目的：递归算法应用		
软件开发工具： Vscode		
<p>1. 实验内容</p> <p>(1) 现有一个有 <math>n</math> 个元素的序列 <math>a=[a_1, a_2, \dots, a_n]</math>，定义这个序列的价值为 <math>\sum_{i=1}^n i \times a_i</math>。空序列的价值为 0。</p> <p>先给你一个长度为 <math>n</math> 的序列 <math>a</math>，求 <math>a</math> 中所有子集价值的异或和，要求子集中元素的相对位置保持不变。</p> <p>(2) 现有一个有 <math>n</math> 个元素的序列 <math>a=[a_1, a_2, \dots, a_n]</math>，定义其价值为 <math>\sum_{i=1}^n a_i \oplus i</math></p> <p>给出这样一个序列，求其所有排列的价值 <math>v_i</math> 的或 <math>v_1   v_2   \dots   v_{n-1}   v_n</math></p> <p>2. 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）</p> <p>(1) 用一个数组记录 1 或 0 表示子串中是否有 <math>a</math> 序列里的数，每个数都有两种可能，两种情况分别调用自身，最后进行递归。对应 1 则将该位置的数复制到另一数组中，最后运算价值。</p> <p>(2) 用递归进行排列，序列中的每个数都有可能出现在各个位置上，所以用一个 for 循环对每个位置上的数与其他位置的数进行交换，然后进行递归，排列到最后一个位置进行计算。</p> <p>3. 测试结果（测试输入，测试输出）</p> <p>(1) 输入：2              1 2              输出：6              输入：3              1 2 3              输出：4</p> <p>(2) 输入：3              1 2 3              输出：6</p> <p>4. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）</p> <p>(1) 结果即子串价值异或</p> <p>(2) 结果即排列组合后的或值</p>		

5. 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>
using namespace std;
int n;
int ans=0;
int ans1=0;
int b[25]={0}; //用 1 或 0 代表子集中是否有这个数
int c[25]={0}; //存子集
void fun(int a[], int i) {
    if(i==n+1) { //从 1 到 n 都由 1 或 0 表示完了
        int num=1;
        for(int j=1; j<=n; j++) {
            if(b[j]==1) {
                c[num++]=a[j]; //b 数组为 1 表示子串中有这个数存到 c 数组里
            }
        }
        ans1=0;
        for(int j=1; j<num; j++) {
            ans1+=j*c[j]; //计算价值
        }
        ans^=ans1;
    }
    else {
        b[i]=1; //每个数字都有两种可能，分别递归
        fun(a, i+1);
        b[i]=0;
        fun(a, i+1);
    }
}
int main() {
    cin>>n;
    int a[25]={0};
    for(int i=1; i<=n; i++) cin>>a[i];
    fun(a, 1);
    cout<<ans;
}
```

(2)

```
#include<iostream>
using namespace std;
int n;
int a[15]={0};
int sum=0;
```

```

int ans=0;

void swap(int& x, int& y) {
    int temp=x;
    x=y;
    y=temp;
} //交换函数

void fun(int i) {
    if(i==n-1) { //一种可能性排列完成，计算结果
        for(int p=0;p<n;p++) {
            sum+=(p+1)^a[p];
        }
        ans|=sum;
        sum=0;
    }
    else{
        for(int j=i;j<n;j++) {
            swap(a[j], a[i]); //后面每个数都有可能放在该位置上，所以每个都进行交换
            fun(i+1);
            swap(a[j], a[i]); //递归后再换回来
        }
    }
}

int main() {
    cin>>n;
    for(int i=0;i<n;i++) {
        cin>>a[i];
    }
    fun(0);
    cout<<ans;
}

```