

数据结构与算法 课程实验报告

学号：202000130143	姓名： 郑凯饶	班级：计科 20.1
实验题目：递归练习		
实验学时：2	实验日期：	
<p>实验目的：</p> <p>1、熟悉开发工具的使用。</p> <p>2、掌握递归的实现思想。</p>		
<p>软件开发环境：</p> <p>Windows 10 家庭中文版 64 位 (10.0, 版本 18363)</p> <p>Dev-C++ IDE</p>		
<p>1. 实验内容</p> <p>1> 求一个序列所有子集价值 $\sum_{i=1}^i i * a_i$ 的异或和。</p> <p>2> 求一个序列所有排序的价值 $\sum_{i=1}^i i^a_i$ 的或。</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> 可以用 $[0, 2^n-1]$ 的二进制数形式表示长度为 n 的序列的所有子集，‘1’ 代表该元素出现。</p> <p>2> 使用递归函数生成全排列，在递归尾部计算价值。</p> <p>3. 测试结果（测试输入，测试输出）</p> <p>在 OJ 平台上成功提交。</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>设计过程中我用的是面向过程的 C 语言设计思想，可以进一步封装，设计成模板类的形式。</p> <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）</p> <p>1></p> <pre> 1. #include<bits/stdc++.h> 2. using namespace std; 3. 4. int ans, A[25]; </pre>		

```

5.
6. int main(){
7.
8.     int n; cin >> n;
9.
10.    for(int i = 1; i <= n; i++) {
11.        cin >> A[i];
12.    }
13.
14.    // 二进制模拟生成子集
15.    for (int i = 0, res, cnt; i <= (1<<n) - 1; i++){
16.        res = 0, cnt = 1;
17.        for (int j = 1; j <= n; j++){
18.            if(i & (1<<(j-1))) res += cnt * A[j], cnt ++;
19.        }
20.        if(!i) ans = res;
21.        else ans ^= res;
22.        // cout << res << " " << ans << '\n';
23.    }
24.    cout << ans << '\n';
25.    return 0;
26.}

```

2>

```

1. #include<bits/stdc++.h>
2. using namespace std;
3.
4. int ans = 0;
5.
6. // 交换法
7. //template<class T>
8. //void permutations(T list[], int k, int m) {
9. // // 生成list[k:m]全排列
10.// if(k == m) {
11.////    copy(list, list+m+1, ostream_iterator<T>(cout, " "));
12.////    cout << '\n';
13.//    int res = 0;
14.//    for (int i = 1; i <= k; i++){
15.//        res += i ^ list[i];
16.//    }
17.//    ans |= res;
18.// }
19.// else {
20.//    for (int i = k; i <= m; i++){
21.//        swap(list[k], list[i]);

```

```
22.// permutations(list, k+1, m);
23.// swap(list[k], list[i]);
24.// }
25.// }
26.//}
27.
28.
29.// DFS: 字典序
30.
31.//void permutations(int list[], int k, int m) {
32.// // 生成list[k:m]全排列
33.// if(k == m) {
34.//     int res = 0;
35.//     for (int i = 1; i <= k; i++){
36.//         res += i ^ list[i];
37.//     }
38.//     ans |= res;
39.// }
40.// else {
41.//     for (int i = k; i <= m; i++){
42.//         swap(list[k], list[i]);
43.//         permutations(list, k+1, m);
44.//         swap(list[k], list[i]);
45.//     }
46.// }
47.//}
48.
49.int main(){
50. int n; cin >> n;
51. int A[11];
52.
53. for (int i = 1; i <= n; i++){
54.     cin >> A[i];
55. }
56. permutations(A, 1, n);
57.
58. cout << ans << '\n';
59.
60. return 0;
61.}
```