武汉大 学计算机学院

本科生实验报告

数据结构实验报告

实验五：递归方法求解 01 背包问题

专 业 名 称 ： 计算机科学与技术

课 程 名 称 ： 数据结构

指 导 教 师 ： 安　　扬

学 生 学 号 ： 2017301500061

学 生 姓 名 ： 彭 思 翔

学 生 班 级 ： 计科二班

上 机 环 境 ： Visual Studio Code

二〇一八 年 11 月

一、实验题目

实验五：递归方法求解 01 背包问题

【问题描述】

设有不同价值，不同重量的物品 n 件，求从这 n 件物品中选取一部分物品的方案，使选中物品的总重量不超过指定的限制重量 W，但选中物品的价值之和最大。注意，每种物品要么被选中，要么不被选中。

【基本要求】

从键盘来输入物品的数量，重量以及限制重量。

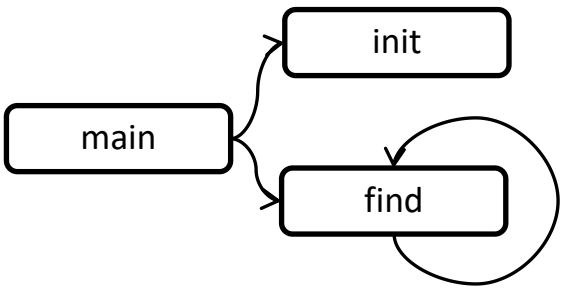
【提示】

采用递归方法求解。

二、实验项目的目的

深入掌握递归设计和求解。

三、实验项目程序结构



四、实验项目中各文件函数功能描述

```
void init();                    //输入函数
void find(int, int);            //递归求解背包问题
```

五、算法描述

【数据结构】

背包和价值：直接简便地用数组存了。

```
double weight[MAXN] = {0}, value[MAXN] = {0};
```

【设计思路】

递归求解：对于当前物品 n 背包剩余空间 w, 在当前物品选或不选中找到最优解递归实现，即递归方程：

$$find(n,w) = \max\{find(n-1,w), find(n-1,w-weight[n]) + value[n]\}$$

时间复杂度 $O(2^n)$ 不到，近似 $O(n \times w)$

```
void find(int x, int w) {
    if (x == 0) {
      ans = max(ans, v);
      return;
    }
    find(x - 1, w);
    if (w >= weight[x]) {
      v += value[x];
      find(x - 1, w - weight[x]);
      v -= value[x];
    }
}
```

六、实验数据和实验结果分析

运行结果良好。

```
Input N:5
Input W:8
Input wights:1 2 3 4 5
Input values:6 7 3 5 9
22.000000
请按任意键继续. . .
```

七、实验体会

这次的 01 背包由于要求用递归实现所以思路非常简单。背包问题是一个经典问题，有更好更快的方法比如动态规划思路，事实上动态规划的转移方程和递归方程是一致的，即

$$f[n][w]=\max\{f[n-1][w],f[n-1][w-weight[n]]+value[n]\}$$

只是动态规划不需要重复计算相同状态的最优值，解决了解的重叠部分的问题，如果上面的递归算法再加上一个二维数组用于记忆状态的最优解就可以达到动态规划的时间复杂度 O(n×w)，甚至更优。另外多重背包、完全背包的问题解法也比较相似。