|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 2 次实验** | | | | | |
| 姓名 |  | 学号 |  | 班级 |  |
| 时间 | 4.20 | 地点 | 软件大楼 028 | | |
| **实验名称** | 用动态规划法求解 0-1 背包问题 | | | | |
| 实验目的 | 通过上机实验，要求掌握动态规划算法的问题描述、算法设计思想、程序设 计。 | | | | |
| 实验原理 | 用动态规划思想设计0-1背包问题 :  01背包问题具有最优子结构性质，即**当前背包可载重W装的东西集合**必然包含 **子问题：背包可载重w<W的东西集合**（当前解的最佳保证子问题的解也是最佳的）  其次，有无后效性：当前背包装不装这个东西与其之前怎么装的无关，只与当前容积有关（子问题的求解过程不会影响当前解的求解过程） | | | | |
| 实验步骤 | 1. 构建数组b[i]用来存放当容量为i时，可以装载的最大价值 2. 从每个输入元素开始遍历，遍历到每个元素有2种选择，装当前元素或不装当前元素，每次取两者最大值记录到b[i] 3. 最终结果为b[W]，代表当重量为W时，背包包含的最大价值 | | | | |
| 关键代码 |  | | | | |
| 测试结果 |  | | | | |
| 实验心得 | 通过这次实验，更加了解了动态规划的原理和性质。要使用动态规划必须满足最优子结构和无后效性这2个性质。动态规划在实践的时候，一般用数组来储存中间结果 | | | | |

附录 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 2 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 4.20 | 地点 | 软件大楼 028 | | |
| **实验名称** | 贪心算法求解背包问题 | | | | |
| 实验目的 | 通过上机实验，要求掌握贪心算法的问题描述、算法设计思想、程序设计。 | | | | |
| 实验原理 | 贪心算法。背包问题具有最优子结构，其中允许装部分内容的背包问题可以用贪心求解 | | | | |
| 实验步骤 | 1. 分析题目发现，要使背包所装的东西价值最高，只需单位质量内装的价值最大即可 2. 由此来设计贪心选择，将每个元素按照单位重量的价值v[i]/w[i]从小到大排序 3. 由最大的元素开始从前往后取，取到装不下为止 4. 将最后那个装不下的取一部分，保证背包刚好装满，并计算其价值即可 | | | | |
| 关键代码 |  | | | | |
| 测试结果 |  | | | | |
| 实验心得 | 这次实验表明，贪心算法比动态规划所利用的资源少，只要确定了贪心选择后，便能很快地解决问题 | | | | |

附录 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 2 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 4.20 | 地点 | 软件大楼 028 | | |
| **实验名称** | 树的最大连通分支问题 | | | | |
| 实验内容 | ★问题描述：给定一棵树T，树中每个顶点u都有一个权w（u），权可以是负数。现在要找到树T的一个连通子图使该子图的权之和最大。  ★算法设计：对于给定的树T，计算树T的最大连通分支。  ★数据输入：由文件input.txt给出输入数据。第1行有1个正整数n，表示树T有n个顶点。树T的顶点编号为1，2…，n。第2行有n个整数，表示n个顶点的权值。接下来的n一1行中，每行有表示树T的一条边的2个整数u，v，表示顶点u与顶点v相连。  ★结果输出：将计算出的最大连通分支的权值输出到文件output.txt。 | | | | |
| 实验原理 | 利用动态规划算法，求出最终的最大权值 | | | | |
| 实验步骤 | 1. 先利用数组v[]储存顶点的权值利用数组tree[][]储存数顶点之间是否相连 ，sum[k]储存以k为头结点的最大权值 2. 设计一个递归函数qiu(k,T),其中，k表示当前节点，T表示父节点，该函数的作用是求出各个sum[] 3. 在qiu函数中，遍历k的每个子节点i（除了k的父节点T），递归调用   qiu（i，k）求出sum[i]，如果sum[i]>0，sum[k]+=sum[i]   1. 最后sum[k]+=v[k] 2. 之后在主函数中对sum排序，最大的sum即为所求 | | | | |
| 关键代码 |  | | | | |
| 测试结果 | 符合要求 | | | | |
| 实验心得 | 这次实验是对动态规划的简单运用，不算难但需要理清楚当前状态与前一状态的关系。 | | | | |

附录 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 2 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 4.20 | 地点 | 软件大楼 028 | | |
| **实验名称** | 会场安排问题 | | | | |
| 实验内容 | ★问题描述：假设要在足够多的会场里安排一批活动，并希望使用尽可能少的会场。设计一个有效的贪心算法进行安排（这个问题实际上是著名的图着色问题。若将每一个活动作为图的一个顶点，不相容活动间用边相连。使相邻顶点着有不同颜色的最小着色数，相应于要找的最小会场数）。  ★算法设计：对于给定的k个待安排的活动，计算使用最少会场的时间表。  ★数据输入：由文件input.txt给出输入数据。第1行有1个正整数k，表示有k个待安排的活动。接下来的k行中，每行有2个正整数，分别表示k个待安排的活动的开始时间和结束时间。时间以0点开始的分钟计。 | | | | |
| 实验原理 | 利用贪心算法，解决会场安排问题 | | | | |
| 实验步骤 | 1. 先将输入的会场按照开始时间顺序从小到大排序 2. 之后从第二个遍历到最后一个会场，当前会场为i 3. 如果在0到i-1的活动j中，有能公用的活动，则gs不变，把活动j   的结束值更新为活动i的结束值   1. 如果活动无法公用，则会场个数gs++ 2. 最终gs即为所求 | | | | |
| 关键代码 |  | | | | |
| 测试结果 | 符合要求 | | | | |
| 实验心得 | 这次实验是对贪心的简单运用，本质是确定贪心策略 | | | | |