|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 3 次实验** | | | | | |
| 姓名 |  | 学号 |  | 班级 |  |
| 时间 | 5.17 | 地点 | 软件大楼 028 | | |
| 实验名称 | 用Dijkstra贪心算法求解单源最短路径问题 | | | | |
| 实验目的 | 通过上机实验，要求掌握 Dijkstra 贪心算法求解单源最短路径问题的问题描述、算法设计思想、程序设计。 | | | | |
| 实验原理 | 根据dijkstra算法，每次选择没选过的，权值最小的点，更新最短距离。 | | | | |
| 实验步骤 | 1. 构建邻接矩阵tu[][]来存储信息，ifuse[i]存储i是否被选择过，currjuli[]赋予初值999，表示当前到各点的最短距离。 2. 根据所给的源点v，和邻接矩阵tu为currjuli[]初始化 3. 根据贪心算法，遍历n次，每次选出没用过的，权值最小的点，更新currjuli[] 4. 最终currjuli[]存的结果即为所求 | | | | |
| 关键代码 |  | | | | |
| 测试结果 | 答案正确 | | | | |
| 实验心得 | 通过这次实验，更加了解了贪心算法的原理和性质。贪心选择每次选择当前的最优解，通过当前最优推出全局最优，dijkstra算法是典型的贪心算法 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 3 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 5.17 | 地点 | 软件大楼 028 | | |
| 实验名称 | 回溯法求解0-1背包 | | | | |
| 实验目的 | 通过上机实验，要求掌握回溯法求解 0-1 背包问题的问题描述、算法设计思想、程序设计。 | | | | |
| 实验原理 | 利用回溯法，遍历每个子集树，得到最终答案。 | | | | |
| 实验步骤 | 1. 从第一个元素开始，根据回溯法，每个元素有去或不去2中选择，遍历这2   种选择，以此遍历每个可能解，并储存他们的当前价值cp和当前重量cw   1. 当cw超过容量c，return 2. 当遍历到最后一个元素时，如果cp超过当前最大bestp，更新解bestx，否则直接返回 | | | | |
| 关键代码 |  | | | | |
| 测试结果 | 答案正确 | | | | |
| 实验心得 | 通过这次实验，更加了解了回溯法的原理和性质。回溯的过程是遍历的过程，只不过会多一些剪枝函数去除肯定不可行的解 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 3 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 5.17 | 地点 | 软件大楼 028 | | |
| 实验名称 | 直线k中值问题 | | | | |
| 实验目的 | 通过上机实验，要求掌握直线k中值问题的问题描述、算法设计思想、程序设计。 | | | | |
| 实验原理 | 贪心：  最多有k个机构，每次从不是机构的城市中选择一个代价最小的城市作为新的机构，更新最小代价。之后在原来的基础上再选择一个代价最小的城市作为新的机构。  这样每次选择都是当前最优解，最终可以得到近似解，无法得到最优解。  动态规划：   1. opt（i，m）是在x1，x2…，xm中恰好设置i个服务机构的最小费用； 2. popt（i，m）是在x1，x2…，xm中恰好设置i个服务机构，且在xm处设置了服务机构的最小费用。 4. 由此可知，所求的最优值为opt=min（opt（i，n）） 5. opt（i，m）和popt（i，m）具有最优子结构性质，且满足如下递归式： | | | | |
| 实验步骤 | 贪心：   1. k次循环，每次选择一个城市作为机构，该城市所造成的代价最少 2. 遍历不是服务站的城市，从中选择代价最小城市作为服务机构处，并且更新每个城市的最小费用A[t][i]。t代表机构数，i代表城市号   动态规划：   1. 读入数据 2. 将popt和opt赋初值（只有一个城市的情况） 3. 对应每次的最多城市数目i，循环计算城市j得到opt[i][j]和popt[i][j]的值 4. 最终答案为opt[i][n]最小的那个值 | | | | |
| 关键代码 | 贪心：      动态规划： | | | | |
| 测试结果 | 贪心：  得到近似解  动态规划：  正确解 | | | | |
| 实验心得 | 通过这次实验，更加了解了贪心和动态规划的区别，贪心有时候只能得到近似解，动态规划确可以得到最优解。但是贪心算法的时间复杂度会比动态规划第一些 | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **算法分析与设计第 3 次实验** | | | | | |
| 姓名 | 陈晨 | 学号 | 201708010719 | 班级 | 计科1707 |
| 时间 | 5.17 | 地点 | 软件大楼 028 | | |
| 实验名称 | 子集和问题 | | | | |
| 实验目的 | 1. 用回溯法求解子集和问题 | | | | |
| 实验原理 | 根据回溯法，每次有取该元素和不取该元素2中选择，遍历这两种选择，直到符合条件或达到尽头 | | | | |
| 实验步骤 | 1. 递归函数传2个参数i（代表当前第几个元素）和curn（代表当前和） 2. 如果当前和currn=c目标和，则更新解x[i]代表是否取第i个元素 3. 如果i>n代表超过元素个数，返回 4. 如果currn+x[i]>c表明已经超过目标和，直接返回，减掉子集 5. 否则递归调用取x[i]和不取x[i]两种情况 | | | | |
| 关键代码 |  | | | | |
| 测试结果 | 答案正确 | | | | |
| 实验心得 | 通过这次实验，更加了解了回溯法的原理和性质。回溯的过程是遍历的过程，只不过会多一些剪枝函数去除肯定不可行的解 | | | | |