**《数据结构与算法》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2021级计算机学院计算机科学与技术卓越2班** | | | **姓名** | **文红兵** |
| **实验题目** | 动态规划算法实践 | | | | | |
| **实验时间** | 2022.11.24 | | **实验地点** | **竹园四栋** | | |
| **实验成绩** |  | | **实验性质** | **□验证性** √**设计性 □综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 实验目的  1. 掌握动态规划算法的基本原理  2. 训练使用动态规划算法设计技术，通过编程解决不同难度问题的实践能力 | | | | | | |
| 二、实验项目内容    实验课题1：  **题目内容：**  7-1 真实的背包故事  **解题思路：**  dp[i][j]:代表前i件物品背包容量为j的最大价值  状态转换方程：dp[ i ][ j ] = min{dp[ i-1 ][ j ],dp[i-1][j-w[i]]+v[i]}  **源代码：**  最少数量：通过标记数组回溯搜索  屏幕截图_20221124_194919  屏幕截图_20221124_194925  屏幕截图_20221124_194932  **时间与空间复杂度分析：**  时间复杂度是O(n^2)  空间复杂度是O(n^2)  实验课题2：  **题目内容：**  7-2 动态规划  **解题思路：**  string dp[maxN][4]：dp[i][j]表示下标i前的字符串分割成j段的反转后的最小字符串  状态转移方程：dp[i][j] = min(dp[i][j],dp[k][j-1]+temp); // k∈[0,i)  **源代码：**  屏幕截图_20221124_194942  屏幕截图_20221124_194952  **时间与空间复杂度分析：**  时间复杂度为O(n^3)  空间复杂度为O(n) | | | | | | |
| 1. **思考题**     0-1背包问题，当商品的重量和价值满足以下哪些条件，就可用比动态规划效率更好的算法解决?描述解题思路并分析时空复杂度。如果没有更好的算法,说明理由（举反例）。 （1）每件商品的价值相同  贪心算法：对重量排序，每次选择重量最小的放进背包，直到背包 放不下，时间复杂度：O(nlogn) 空间复杂度：O(1) （2）每件商品的重量相同  类似1：贪心算法：对价值排序，每次选择价值最大的放进背包，直到背包放不下，时间复杂度：O(nlogn) 空间复杂度：O(1) （3）每件商品的性价比相同 （性价比 = 价值/重量）  没有更好的算法解决  如背包容量:5  商品：(1,1),(1,1),(1,1),(1,1),(5,5) （4）重量越大，性价比越低  没有更好的算法解决  如背包容量:5  商品：(1,10),(2,18),(3,24),(4,28),(5,30) （5）重量越大，价值越小  贪心算法：对重量排序，每次选择重量最小的，直到装不下  时间复杂度：O(nlogn) 空间复杂度：O(1) | | | | | | |
|  | | | | | | |