天津大学本科生实验报告专用纸

学院 智能与计算学部 年级 2018级 专业 软件工程 班级 6 姓名 王传安 学号 3018216301 课程名称 算法设计与分析 实验日期 2019/10/30

同组实验者 无 成绩

|  |
| --- |
| 1. 实验目标：   利用动态规划算法求解旅行商问题。   1. 实验内容：   旅行家要旅行5个城市，要求各个城市经历且仅经历一次然后回到出发城市，并要求所走的行程最短，其代价矩阵如下（INF表达不可达）。  试求出最小代价，并输出对应的路径。     1. 实验步骤：   根据动态规划算法的原理，我们需要找到要解决的问题的子问题。  题目要求从0出发，经过[1, 2, 3, 4] = S，这几个城市，然后回到0，子问题为从0出发到i（i属于S），然后再从i出发到S-i，然后回到0，使得花费最小。  设置一个二维动态规划表dp，定义符号{1，2，3，4}表示经过[1, 2, 3, 4]这几个城市然后回到0.  所以目标即为求dp[0][{1, 2, 3, 4}]。将{1, 2, 3, 4}定义为二进制1111，也就是十进制的15，即求dp[0][15]。 |

天津大学本科生实验报告专用纸

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dp[0][{1, 2, 3, 4}] = min{C01 + dp[1][{2, 3, 4}] + min{C02 + dp[2][{1, 3, 4}] + min{C03 + dp[3][{1, 2, 4}] + min{C04 + dp[4][{1, 2, 3}]  以此类推，得到整个dp表。  dp表有5行（5个城市），1 << (5 – 1)列。   |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | {} | {1} | {2} | {1,2} | {3} | {1,3} | {2,3} | {1,2,3} | {4} | |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 0 | -1 | 6 | -1 | -1 | 16 | 21 | -1 | 18 | 18 | | 1 | 3 | -1 | -1 | -1 | 18 | -1 | 15 | -1 | 14 | | 2 | -1 | 6 | -1 | -1 | 12 | 17 | -1 | -1 | 12 | | 3 | 8 | 13 | -1 | 10 | -1 | -1 | -1 | -1 | 29 | | 4 | 9 | 8 | -1 | 9 | 28 | 23 | 15 | 20 | -1 |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | {1,4} | {2,4} | {1,2,4} | {3,4} | {1,3,4} | {2,3,4} | {1,2,3,4} | |  | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | 0 | 17 | -1 | 18 | 37 | 32 | 24 | 23 | | 1 | -1 | 15 | -1 | 33 | -1 | 20 | -1 | | 2 | 11 | -1 | -1 | 31 | 26 | -1 | -1 | | 3 | 24 | 16 | 15 | -1 | -1 | -1 | -1 | | 4 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |   TSP函数：先初始化dp表的第一列（直接从邻接矩阵中获得）  用((j >> (i – 1)) & 1) == 1来判断集合j中是否含i结点。在遍历的过程中，如果集合j含i结点，退出，增加一个1 <= k < N的遍历，如果达不到k结点，退出。  最后dp[0][15]即为最小值 |

|  |
| --- |
| getPath获得对应最小值的路径： |
| **教师签字：**  **年 月 日** |