

天津大学本科实验报告专用纸

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_____

一、 实验目标：

利用动态规划算法求解旅行商问题。

二、 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。

<

```

void TSP() {
    /*
    初始化dp[i][0]
    */
    for (int i = 0; i < N; i++) {
        dp[i][0] = g[i][0];
    }

    for (int j = 1; j < M; j++) {
        for (int i = 0; i < N; i++) {
            dp[i][j] = INF;
            //如果j中包含结点i,退出
            if (((j >> (i - 1)) & 1) == 1) {
                continue;
            }
            //如果不能到达k,退出
            for (int k = 1; k < N; k++) {
                if (((j >> (k - 1)) & 1) == 0) {
                    continue;
                }
                if (dp[i][j] > g[i][k] + dp[k][j ^ (1 << (k - 1))]) {
                    dp[i][j] = g[i][k] + dp[k][j ^ (1 << (k - 1))];
                }
            }
        }
    }
}

```

getPath 获得对应最小值的路径:

```

void getPath() {
    //标记访问数组
    bool visited[N] = { false };
    //前驱节点编号
    int pioneer = 0, min = INF, S = M - 1, temp;
    //把起点结点编号加入容器
    path.push_back(0);
    while (!isVisited(visited)) {
        for (int i = 1; i < N; i++) {
            if (visited[i] == false && (S & (1 << (i - 1))) != 0) {
                if (min > g[i][pioneer] + dp[i][(S ^ (1 << (i - 1)))]) {
                    min = g[i][pioneer] + dp[i][(S ^ (1 << (i - 1)))];
                    temp = i;
                }
            }
        }
        pioneer = temp;
        path.push_back(pioneer);
        visited[pioneer] = true;
        S = S ^ (1 << (pioneer - 1));
        min = INF;
    }
}

```

教师签字:

年 月 日

