**西 南 石 油 大 学 实 验 报 告**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **课程** | **算法分析与设计** | **实验项目** | | **贪心算法设计与实现** | | | |
| **专业班级** | **计算机科学与技术2022级** | | **姓名** | | **王磊** | **学号** | **202231060435** |

### 1、实验内容

（1）实验步骤

1. 了解贪心算法的基本原理和应用场景。
2. 选择一个具体的最短路径问题作为实验对象，例如Dijkstra算法。
3. 编写贪心算法的伪代码和实现代码。
4. 设计测试数据并进行实验测试。
5. 分析实验结果，比较贪心算法的性能。
6. 算法原理

贪心算法是一种在每一步选择中都采取当前最优选择的算法。它通过逐步构建解决方案，并保证每一步都选择局部最优解，从而期望最终能达到全局最优。对于最短路径问题，Dijkstra算法是一个经典的贪心算法。

Dijkstra算法的主要思想是从起点开始，逐步扩展到其他节点，每次选择到达当前节点集合中最短路径的未访问节点，更新其邻接节点的路径长度。

1. 测试数据

顶点：6

边：9

边的距离：

1 -> 2 :1

1 -> 3 :12

2 -> 3 :9

2 -> 4 :3

3 -> 5 :5

4 -> 3 :4

4 -> 5 :13

4 -> 6 :15

5 -> 6 :4

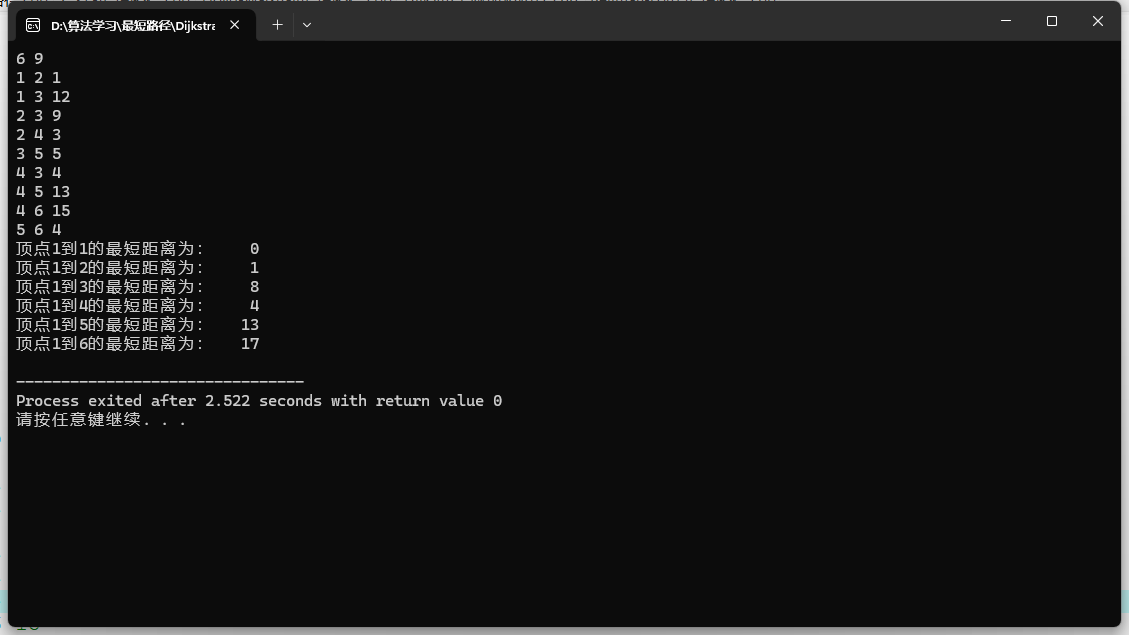
1. 程序主要流程
2. 初始化：设置起点到自身的距离为0，其他节点的距离为∞。
3. 选取当前未访问节点中距离最小的节点，标记为已访问。
4. 更新该节点的邻接节点的距离，如果通过该节点到邻接节点的距离更短，则更新邻接节点的距离。
5. 重复步骤2和3，直到所有节点都被访问过。
6. 主要模块功能

* 初始化模块：初始化节点距离和优先队列。
* 选择节点模块：从优先队列中选取当前最短路径的未访问节点。
* 更新距离模块：更新邻接节点的最短路径距离。
* 主循环模块：实现主算法循环，直到所有节点都被访问。

1. 数据结构设计

* 使用优先队列（最小堆）来高效选择当前最短路径的节点。
* 使用字典存储每个节点的当前最短距离。
* 使用集合来记录已访问的节点。

1. 运行结果（截图）



（8）算法时间复杂度

Dijkstra算法的时间复杂度主要取决于使用的数据结构。使用最小堆时，其时间复杂度为O((V+E)logV)，其中V是节点数，E是边数。

### 1、小结

（自己对于实验的结论、认识、心得/体会等）

通过此次实验，我了解了贪心算法在解决最短路径问题中的应用，并通过编程实现了Dijkstra算法。实验结果表明，贪心算法能够有效地求解较大图的最短路径问题。通过实验，我加深了对贪心算法的理解，并掌握了如何分析和优化算法的性能。

### 附录：源代码

