重庆大学本科学生实验项目任务书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验题目 | 图算法实践 | | | |
| 实验时间 | 2022.12.8 | 实验地点 | | DS3402 |
| 实验性质 | □验证性 √设计性 □综合性 | | | |
| 实验目的  1. 掌握图的存储结构与基本操作  2. 训练使用经典的图算法，通过编程解决不同难度问题的实践能力 | | | | |
| 实验内容   1. 图的基本数据结构与基本操作，经典图算法 2. 在线编程解决2道不同难度的编程题，共40分   3． 在线编程时间（答题时长）150分钟，从开始答题时刻起算  4． 在线程序测评平台采用PTA (https://pintia.cn/)  5． 每人必须独立完成编程，可查阅教科书、PPT等资料，不得相互抄袭以及抄袭网上已有的程序  6． 实验课结束后，会对所有程序进行查重，如检测出有抄袭的程序，成绩计零分处理  注意：最后提交完整的实验报告，包括对每道题的算法思路、代码描述、复杂度分析等内容并回答思考题。 | | | | |
| 参考资料：   * Data Structures and Algorithm Analysis (C++ Version) Clifford A. Shaffer   + Introduction to Algorithms, 3rd Edition, MIT Press, T.H. Cormen, et al.   + 《数据结构（ C 语言版）》，严蔚敏，吴伟民编著，清华大学出版社 | | | | |
| 任务下达日期 2022 年 12月 1日 | | | 完成日期 2022 年12 月 10 日 | |

说明：学院、专业、年级均填全称，如：计算机学院、计算机科学与技术、2021。

**《数据结构与算法》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **21级计算机科学与技术(卓越)1班** | | | **姓名** | **罗帆靖** |
| **实验题目** | 图算法实践 | | | | | |
| **实验时间** | **2022.12.8** | | **实验地点** | **DS3402** | | |
| **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. 掌握图的存储结构与基本操作  2. 训练使用经典的图算法，通过编程解决不同难度问题的实践能力 | | | | | | |
| 二、实验项目内容  注：每道题按下面的格式分别描述  实验课题1：  题目内容： 生化危机  解题思路： 利用邻接表储存图，此图中加入起点以及两个端点都为安全城市的边。首先判断终点是否安全，其次在图中在起点开始对图做一个DFS，查看有无到终点的连通即可。  源代码（格式正确，对关键步骤加注释）：      时间与空间复杂度分析:  使用图的DSF算法，每个点需遍历一次，时间复杂度O(m);  采用邻接表方式存储，每个定点需存储其邻接边。采用队列遍历方式。空间复杂度为O(m+k);  实验课题2：  题目内容：大众情人  解题思路： 采用floyd算法求出各点之间的最短路径，再遍历一次找到一个点的另一个点性别不同于其且最长的路径（倒数即为其异性缘）。最后找出大众情人（异性缘最大）的点，利用向量顺序存储并输出以达到有序输出的结果。  源代码（格式正确，对关键步骤加注释）：            时间与空间复杂度分析:  采用floyd算法求最短路径，时间复杂度为O（N^3）;  利用邻接表存储图，向量存储性别，空间复杂度为O(N^2); | | | | | | |
| 三、思考题  对于无边权重的无向图，可以用广度优先遍历(BFS)算法求从起点到其它所有结点的最短路径，并且使用的FIFO队列的长度为O(|V|),即结点的数量。而有权重的图，使用Dijsktra算法求最短路径，通常需要使用长度为O(|E|)的优先队列。造成这一区别的主要原因是什么？如果要让Dijsktra算法也只使用O(|V|)长度的优先队列，该如何处理？  主要原因是，Dijsktra算法采用是否已经找到最短路径为标记的根据，因此可能会将同一节点多次入队，长度可能达到O(E)；然而，BFS采用是否被访问作为标记的依据，每个结点必定只会入队一次，因此长度为O(V);  若Dijsktra算法也只使用O(V)长度的优先队列，则在每次节点入队的时刻，遍历整个队列，查看是否有相同节点，若有则更新节点到起点的最短路径；若没有则入队。 | | | | | | |