

创邻科技-条件环路子图算法设计

CCF 竞赛方案

1. 赛题介绍

1.1. 赛题概述

本次竞赛针对参赛者开发的图算法在正确性、稳定性、性能和内存开销几个维度进行综合评估。参赛者通过下载提供的数据集，利用 Galaxybase 图平台建图，并在图上进行算法设计，用标准输出结果进行算法正确性验证。

1.2. 赛题内容

给定一张含有 m 条边的有向图。每个查询会输入一个起始点，要输出从这个起始点出发，符合给定条件的所有环路组成的子图。条件环路子图的计算规则如下：

- 1、环路的含义是，从起始点出发，根据有向边的传递方向，最终回到起始点。在任何一条环路上的边最多只能经过一次（而点可以重复经过，不限次数）；
- 2、不同环路之间的边可以重合，最终输出的是不同环路共同组成的边集合（边集合需要按照边自增编号 e_id 进行升序排序）；
- 3、要求环路的路径长度（即所包含的边数）大于等于 d_min ，小于等于 d_max ；
- 4、要求环路中的任意一条边的时间属性，和其前驱边的时间属性的差值在 d_day 之内（包含 d_day ），环路中的第一条边的时间属性没有限制。

2. 数据集介绍

2.1. 数据集说明

参赛者通过下载数据集进行图构建，并在图上进行算法设计。其中 `transfer.csv` 用来建图使用；`sample_input.csv` 作为样本输入；`sample_output.csv` 作为样本输出。

2.2. 数据集获取方式

数据集下载链接：

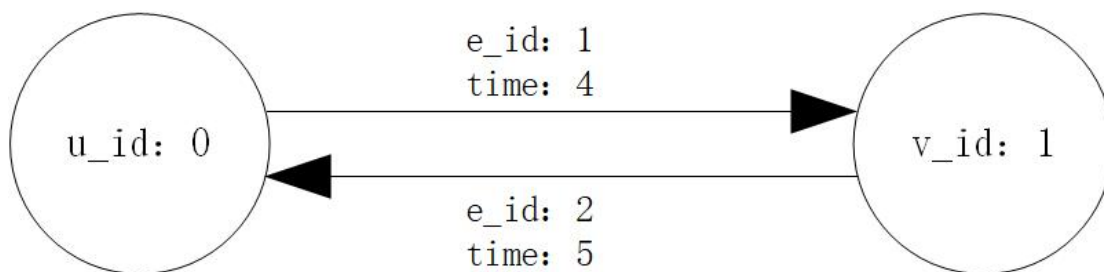
https://galaxybase.com/public/activity/task2_data.zip

2.3. 数据集文件介绍

1、图数据文件../sampleData/transfer.csv，其中第一列表示边自增编号 e_id，第二列表示起点编号 u_id，第三列表示终点编号 v_id，第四列表示时间属性 time，每一列之间用逗号分隔，例如：

e_id	u_id	v_id	time
1	0	1	4
2	1	0	5

上述数据表示一共有两条边，第一条边的编号为 1，第二条边的编号为 2。图中共有两个点，起点 0 和终点 1，这两个点之间有两条边，分别为时间点 4 创建的边和时间点 5 创建的边。



2、查询样例输入文件../sampleData/sample_input.csv，其中第一列表示起点编号 start_id，第二列表示时间间隔 d_day，第三列表示最短路径长度 d_min，第四列表示最长路径长度 d_max，每一列之间用逗号分隔，例如：

start_id	d_day	d_min	d_max
0	1	2	5
1	2	3	5

上述数据表示一共有两个查询，其中第一个查询起点为 0，时间间隔为 1，路径长度在 [2, 5] 之间的合法连通图。

3、查询样例输出文件../sampleData/sample_output.csv, 算法运行后的结果输出至../sampleData/output.csv 文件中, 你可以通过观察的方式进行比对, 也可以自己写比对程序进行比对。输出格式如下, 每一个查询输出两行, 第一行首先输出 case 1 表示这是第一个查询, 中间一个空格, 然后是一个整数表示条件环路子图一共有几条边, 第二行多个用逗号分隔的整数, 表示升序输出的边编号 e_id, 例如:

```
case1 2
```

```
1,2
```

```
case2 0
```

上述数据表示一共有两个案例, 其中第一个案例的条件环路子图边总数为 2, 边编号分别是 1 和 2。第二个案例没有解, 边总数为 0。

关于 sample_input.csv 与 sample_output.csv 的具体使用方法请参见本文 3.4 节“算法编写”详细内容。

2.4. 数据集范围

- 1、边总数, 正整数, $m \approx 200w$ 。
- 2、路径长度, 正整数, $2 \leq d_{\min} \leq d_{\max} \leq 8$ 。
- 3、时间间隔, 非负整数, $0 \leq d_{\text{day}} \leq 20w$ 。
- 4、边自增编号属性, 正整数, 连续自增, $1 \leq e_{\text{id}} \leq m$ 。
- 5、起点编号和终点编号, $1 \leq u_{\text{id}}, v_{\text{id}}, \text{start_id} \leq m$ 。
- 6、时间属性, 非负整数, $0 \leq \text{time} \leq 20w$ 。

3. 答题流程

3.1. 平台搭建

如需自行搭建 Galaxybase 图平台，具体平台搭建流程可参见《附件 3-GalaxyBase 安装部署配置指南》，也可参见《Galaxybase 开发者版本文档》中“安装部署配置”全部章节。

相关内容文档链接：

<https://www.galaxybase.com/document?file=dev&docid=27>

如不具备搭建 Galaxybase 图平台条件可在比赛报名表中申请远程环境。

3.2. 数据建模

1、创建一个点，点类型名称为 vertex，点主键唯一标识为 id。



创建点

点类型名称: vertex

外部唯一标识: id

样式:

请选择颜色（十六进制色） #448DFF

选择图标

2、创建一条边，边类型名称为 edge，起止点都是 vertex，方向有向。有两个属性，一个是边自增编号属性 e_id，一个是时间属性 time。

 **编辑边（边类型相同，属性也相同）** 

边类型名称：

edge

起始点类型： 终止点类型： 方向

vertex vertex 有向

样式：

 请选择颜色（十六进制色）
#448DFF

属性：

单击“+”添加属性 

eld	INT		
time	INT		

3、图模型如下图所示。



3.3. 数据建图

1、选择数据../sampleData/transfer.csv。

文件格式:	分隔符:	文件编码:	包围字符
CSV	,	UTF-8	"
<input type="checkbox"/> 有标题			
column_1	column_2	column_3	column_4
格式检查: STRING	STRING	STRING	STRING
1	1073	1073	80115
2	800	964	80015
3	966	971	80015

2、建立映射关系。



3、数据导入。

transfer.csv

图名称: Graph

加载文件: [查看历史加载记录](#)

数据文件	数据映射 (点/边)	加载类型	加载状态
上传目录:transfer.csv		自动追加	未加载

选择加载类型: ☒ 增量 ☐ 覆盖

加载

本次加载日志:

加载起始时间: Wed Jun 16 18:50:21 CST 2021, 加载终止时间: Wed Jun 16 18:50:39 CST 2021
加载类型: buildGraph
加载最终状态: LoadSuccess
数据量: 1_上传目录:transfer.csv: 总行数: 1996234 错误行数: 0

4、导入成功后显示如下图。



3.4. 算法编写

编写 Java API 程序，算法执行的入口方法是 CCFGraphAlgo2021.java 类的 main 方法，读入样本输入../sampleData/sample_input.csv 文件作为样本查询输入，执行完算法的结果输出至../sampleData/output.csv 文件中，同../sampleData/sample_output.csv 样本输出文件进行比对。

下面提供一个具体案例：

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class CCFGraphAlgo2021 {
    public static void main(String[] args) throws IOException {
        // 样本输入路径，输入文件名 sample_input.txt
        String inputFilePath = args[0];
        // 算法输出路径，输出文件名 output.txt
        String outputFilePath = args[1];
```



```
Scanner sc = new Scanner(new File(inputFilePath));
FileWriter fw = new FileWriter(outputFilePath);
int ca = 0;
while (sc.hasNext()) {
    long startTime = System.currentTimeMillis();
    String[] input = sc.nextLine().split(",");
    int e_id = Integer.parseInt(input[0]);
    int d_day = Integer.parseInt(input[1]);
    int d_min = Integer.parseInt(input[2]);
    int d_max = Integer.parseInt(input[3]);

    // 执行算法（举例）
    execAlgo(e_id, d_day, d_min, d_max, fw);

    long endTime = System.currentTimeMillis();
    System.out.println("case" + ++ca + " cost = " + (endTime -
startTime) + " ms");
}
fw.close();
}

// 选手自写图算法模块
private static void execAlgo(int e_id, int d_day, int d_min, int
d_max, FileWriter fw) throws IOException {
    /*
    算法实现
    **/
```

```
// 结果输出

fw.write("case1 2\n");

fw.write("1,2\n");

}

}
```

关于 Java API 的使用详情请参考以下文档链接：

<https://galaxybase.com/document?file=dev&docid=25>

3.5. 算法验证

工作人员会在 8 核 16G 的机器上，把标准输入 sample_input.csv 放入 ../sampleData 文件夹下，并导入数据集 ../sampleData/transfer.csv。导入完成后，工作人员会执行 CCFGraphAlgo2021.java 类的 main 方法进行测试，读入标准输入进行输入，算法在 1 小时内跑出结果后，将结果输出至 ../sample/output.csv 文件中，然后工作人员会把这个结果和标准输出进行比对，若正确则获得该查询的得分。

4. 提交项说明

选手提供的提交物的文档结构示意图如下：



文件名命名规则：【选手姓名】+ “_” + 【任务全名】

具体文件介绍：

CodeData：相关要求输出程序源代码存放位置。

README.md：算法设计说明文档。

5. 评测结果

工作人员将在至多 5 个工作日内给出评测结果，评测结果可参考下图。

案例编号	案例测试结果	用时/错误日志
1	正确	20 分钟 12 秒
2	正确	10 秒
3	正确	1 秒
4	错误	运行超时
5	错误	内存溢出，错误日志
6	错误	运行超时
7	错误	运行超时
8	错误	空指针异常，错误日志
9	错误	数组越界，错误日志
10	错误	除零异常，错误日志

案例编号	案例测试结果	用时/错误日志
1	程序无法运行	例如：找不到对应 xxx 依赖；根据说明无法执行程序。

6. 评分标准

评分将在比赛结束后进行最终统计计算。

序号	评测指标	分值	评测指标说明	评分标准
1	正确性	30	查询是否正确。	10 个查询总共 30 分，其中每个查询正确得 3 分，不正确得 0 分。
2	性能	50	正确性通过的才有对应的性能得分。	10 个查询总共 50 分，其中每个查询最高 5 分，最低 1 分，按运行时间排名顺序决定。
3	说明文档	20	完备性、可读性和格式规范等。	根据文档的完备性、可读性和格式规范进行评分。
	合计	100		