



HUAWEI Code Craft 2016

未来网络·寻路

初赛题目介绍

2016.03.04

前言

赛题源自“未来网络”业务发放中的路由计算问题。算路问题属于基础算法问题，在图论、网络、交通等各个方面均有着广泛的研究与运用，里面不乏一些经典的算法，例如最短路径中的广度优先搜索，Dijkstra算法等。网络算路问题的更优算法实现对于网络资源高效配置具有重要价值。

1 问题定义

给定一个带权重的有向图 $G=(V,E)$ ， V 为顶点集， E 为有向边集，每一条有向边均有一个权重。对于给定的顶点 s 、 t ，以及 V 的子集 V' ，寻找从 s 到 t 的不成环有向路径 P ，使得 P 经过 V' 中所有的顶点(对经过 V' 中节点的顺序不做要求)。

若不存在这样的有向路径 P ，则输出无解，程序运行时间越短，则视为结果越优；若存在这样的有向路径 P ，则输出所得到的路径，路径的权重越小，则视为结果越优，在输出路径权重一样的前提下，程序运行时间越短，则视为结果越优。

说明：

- 1) 图中所有权重均为 $[1, 20]$ 内的整数；
- 2) 任一有向边的起点不等于终点；
- 3) 连接顶点 A 至顶点 B 的有向边可能超过一条，其权重可能一样，也可能不一样；
- 4) 该有向图的顶点不会超过 600 个，每个顶点出度(以该点为起点的有向边的数量)不超过 8；
- 5) V' 中元素个数不超过 50；
- 6) 从 s 到 t 的不成环有向路径 P 是指， P 为由一系列有向边组成的从 s 至 t 的有向连通路径，且不允许重复经过任一节点；



7) 路径的权重是指所有组成该路径的所有有向边的权重之和。

2 输入与输出

输入文件格式

以两个.csv 文件(csv 是以逗号为分隔符的文本文件)给出输入数据, 一个为图的数据(G), 一个为需要计算的路径信息(s,t,V')。文件每行以换行符(ASCII'\n'即 0x0a)为结尾。

1) 图的数据中, 每一行包含如下的信息:

LinkID,SourceID,DestinationID,Cost

其中, LinkID 为该有向边的索引, SourceID 为该有向边的起始顶点的索引, DestinationID 为该有向边的终止顶点的索引, Cost 为该有向边的权重。顶点与有向边的索引均从 0 开始编号(不一定连续, 但用例保证索引不重复)。

2) 路径信息中, 只有一行如下数据:

SourceID,DestinationID,IncludingSet

其中, SourceID 为该路径的起点, DestinationID 为该路径的终点, IncludingSet 表示必须经过的顶点集合 V', 其中不同的顶点索引之间用'|'分割。

输出文件格式

输出文件同样为一个.csv 文件。

1) 如果该测试用例存在满足要求的有向路径 P, 则按 P 经过的有向边顺序, 依次输出有向边的索引, 索引之间用'|'分割;

2) 如果该测试用例不存在满足要求的有向路径 P, 则输出两个字符 NA;

3) 只允许输出最多一条有向路径。

3 单个用例的评分机制

有解用例的排名机制

按下面流程对参赛者结果进行排名:

Step1: 对于提交的结果, 进行合法性检验(详见题目描述);

Step2: 程序运行时间不得超过 10s;



若不满足上述的结果则本用例得分为 0；

Step3: 计算提交的路径的权重，权重越小，排名越优；

Step4: 在权重相同的结果里，用程序运行时间进行排名，时间越短，排名越优。

无解用例的排名机制

按下列流程对参赛者结果进行排名：

Step1: 对于提交的结果，验证是否识别出该用例无解，若无法识别或者算法运行时间超 10s，则本用例得分为 0；

Step2: 用程序的运行时间进行排名，时间越短，排名越优。

单个用例的评分标准如下：

根据上面排名流程得到的排名，使用标准分计分(排名第一的提交者为 100 分)。

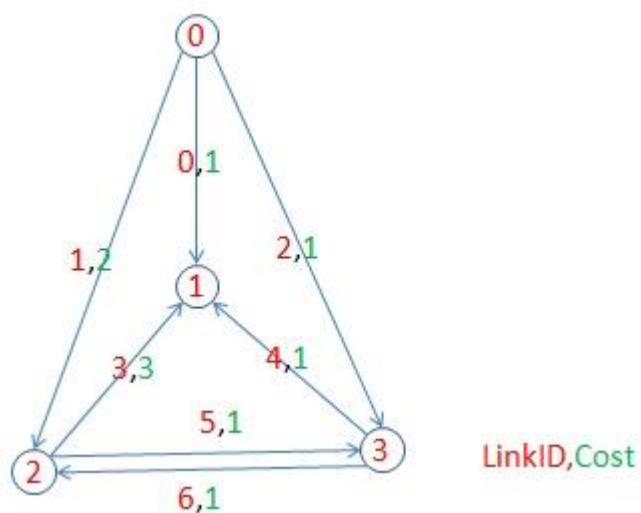
若所有人均未得到正确结果，则所有人均得分为 0。

4 最终得分机制

平台会使用 N 个测试用例判题，该 N 个测试用例分为初级、中级、高级三个等级，参赛者对于每个测试用例都会得到一个百分制分数，使用加权平均分(初级权重为 0.2，中级权重为 0.3，高级权重为 0.5)作为该参赛者的最终得分。

特别说明：在比赛初期，平台只放出初级、中级的测试用例，故此时满分为 50 分，在比赛后期，才会放出高级测试用例（具体发放时间会在网站公告通知），此时满分才为 100 分，请各位参赛者注意。

5 简单用例说明



在如上图所示的有向图中，我们会得到下面的有向图信息：

0,0,1,1

1,0,2,2

2,0,3,1

3,2,1,3

4,3,1,1

5,2,3,1

6,3,2,1

如果此时需要寻找从 0 到 1 的路径，且必须经过顶点 2 和 3，我们会得到如下的路径信息：

0,1,2|3

对于该用例，可以找到如下两条可行路径：

1|5|4

2|6|3

由于第一条路径的权重为 4，第二条路径的权重为 5，所以此时最优解应该 1|5|4。

< end >