北京郵電大學

实验报告

实验名称: 求有向无环图中最长路径

班级 : 2019211309 姓名 : 陈悦 学号 : 2019211413 分工 : 文档

班级 : 2019211309 姓名 : 马晓亮 学号 : 2019211400 分工 : 代码

2020 年 *12* 月 *6* 日

一　需求分析

本次实验需要设计一个程序，程序实现了计算一个有向无环图中最远的两点的距离，并且，本程序会打印其中一对的路径。

输入样例:

6 8

0 1 3

0 2 1

0 3 2

1 4 4

2 4 7

3 1 1

3 5 5

5 4 6

输出样例:

The longest disatance is: 13

The longest_path is:

0 » 3 » 5 » 4 » end

二　概要设计

由于我们输入的图为有向无环图 G，所以 G 中会存在一组源点和汇点，则最远两点的距离必定是从源点到汇点的距离。所以我们采用了动态规划的思想，即维护一个 dis, 表示一个顶点最远源点的距离，我们对 G 进行拓扑排序，按照拓扑排序的顺序对 dis 更新。并且维护每个顶点的 father，表示其 dis 的前驱顶点，方便输出路径，所以我们的数据结构如下

邻接链表:

1. weight 边权
2. self 目标顶点

3. adjvex 当前顶点

4. next 下一条边

结点结构:

1. data 顶点保存的数据

2. indegree 入度

3. longest 到 m 顶点的最长距离

4. father 最长距离路径上的父亲节点

5. firstAdj 顶点的第一条边

图类:

1. vertex 顶点列表，用 vector 存储

2. ver_num 顶点的个数

3. edge_num 边的个数

三　详细设计

Algorithm 1 longest_path

```
初始化队列 topo_node
将所有入度为 0 的结点加入到 topo_node 中
while topo_node 不为空 do
    从 topo_node 中取出 front() 保存在 temp 中并弹出
    for 从 temp 出发的所有边 e do
        将 e 的终点的顶点入度减 1
        if e 的终点的入度为 0 then
            将 e 的终点加入到 topo_node 中
        end if
        if temp 的 longest + e->weight > e 的终点的 longest then
            e 的终点的 longest = temp 的 longest + e->weight
            更新 e 的终点的 father 为 temp
        end if
    end for
end while
```

Algorithm 2 create_graph

```
输入 n、m 表示顶点的个数和边的个数
初始化 n 个顶点
for 对于 m 条边 do
    输入 x, y, w 表示起点，终点，边权
    利用 x,y,w 初始化边
    将边插入到邻接表中
end for
```

四　调试分析报告

在我们设计的算法中，每个点仅能进入一次队列，且每个边只能被删除一次，所以算法复杂度只和边与点的个数有关。

所以算法复杂度为 $O(n+e)$

五　用户使用说明

用户可以使用 IDE 或者手动编译源代码 main.cpp，获得可执行文件。

笔者使用的 gcc 版本为 8.1.0

在使用时，用户需要先输入两个数字 n,m 表示顶点数与边数，再依次输入每一条边的具体数值

得到的结果中包括最长距离与一条表示最长距离的路径

六　测试结果

我们使用的测试样例如下图所示

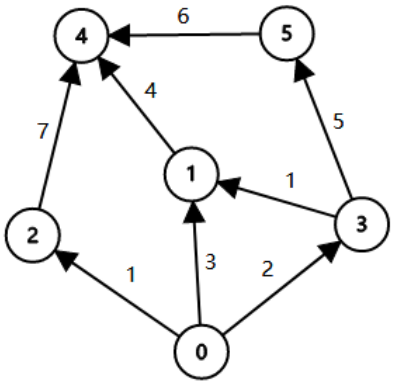

图 1: graph

输入为 6 8

0 1 3

0 2 1

0 3 2

1 4 4

2 4 7

3 1 1

3 5 5

5 4 6

结果为

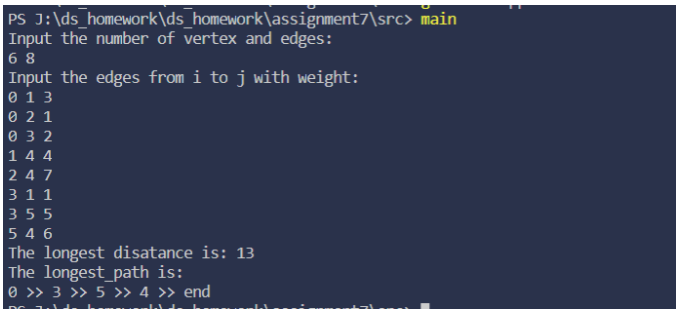

图 2: 测试结果