**深 圳 大 学 实 验 报 告**

**课程名称： 算法设计与分析**

**实验项目名称： 图论：桥问题**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 马里佳**

**报告人： 钟善扬 学号： 2017303031 班级： 03**

**实验时间： 2020/06/30**

**实验报告提交时间： 2020/07/05**

**教务处制**

### 一、实验目的：

* + 1. 掌握图的连通性。
    2. 掌握并查集的基本原理和应用。

### 二、内容：

**1. 桥的定义**

在图论中，一条边被称为“桥”代表这条边一旦被删除，这张图的连通块数量会增加。等价地说，一条边是一座桥当且仅当这条边不在任何环上。一张图可以有零或多座桥。

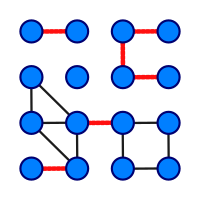
 

图 1 没有桥的无向连通图 图 2 这是有16个顶点和6个桥的图

（桥以红色线段标示）

**2. 求解问题**

找出一个无向图中所有的桥。

1. **实验过程**

**（1）利用基准算法求取图中桥的数目**

**算法思路：**

使用bfs或者dfs遍历每一条边（u，v），看看从u出发，不经过（u，v），是否能到达v。如果可以，则这条边不是桥。如果不可以，那么这条边就是桥。

伪代码如下：

For every edge (u, v), do following

a) Remove (u, v) from graph

b) See if the graph remains connected (We can use BFS or DFS)

c) Add (u, v) back to the graph.

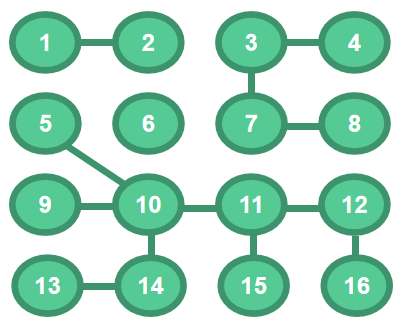
**（2）应用并查集设计一个比基准算法更高效的算法**

设计了LCA+并查集缩环成点算法。思路：

LCA，即最低公共祖先（Lowest Common ancient），有最低公共祖先的两个点一定连通了，如果再连接这两个点，将会形成环。环中的边都不是桥。

首先通过找到生成树。求最小生成树的方法是克鲁斯卡尔和普里姆算法，比如说普里姆算法：对于任意两个点，如果两个点在一个集合中，我们就不做处理，如果不在一个集合中，我们就把这两个点连接起来,这里查看是否属于同一集合实际上也用到了并查集。

由于桥必在生成树上，而我们再加入边的话生成树一定会出现环，环上的边都不是桥，我们就可以把这个环缩成“点”（并查集的“并”）。



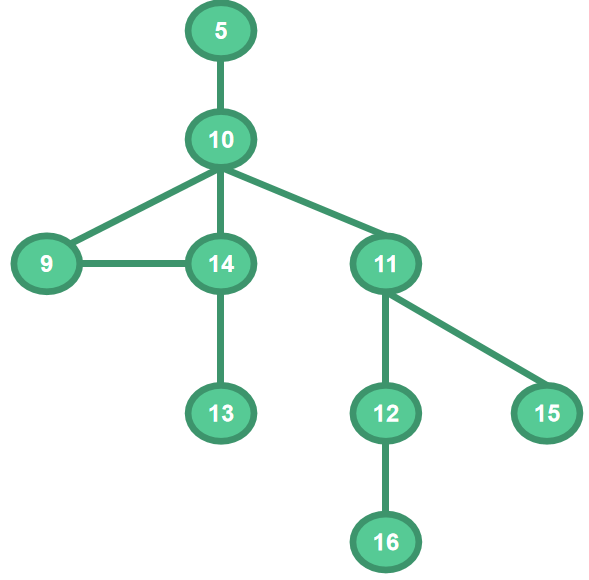
其次，很明显，最小生成树的生成过程要求桥两端的点属于不同的集合，因此，桥必须是最小生成树的一部分。 现在，我们首先假设这棵树的每个边缘都是一座桥梁。

那么不在最小生成树上的边一定为环边，因为最小生成树上面的点都是两两可达的，你再加一条肯定多余了。所以接下来我们一个个把这些环边加进去，然后再缩环成点，就是在一个环上的我们都给他并起来，让他们有同一个代表，也就是他们的最低公共祖先。

下面以上面的例子演示我的算法思想：

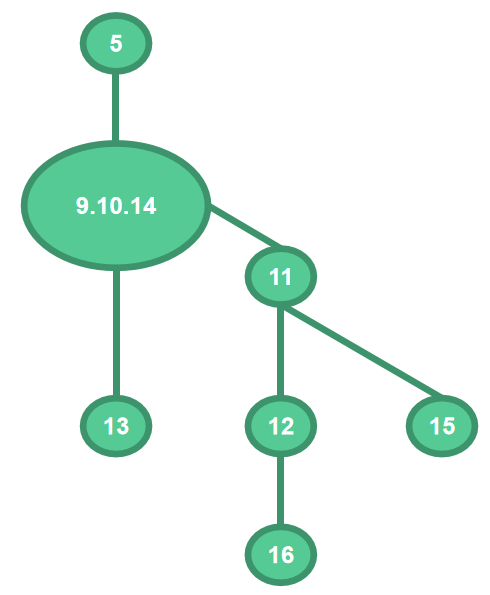
生成树外的边有：（9，14），（15,16），（5,9）

* （9，14）

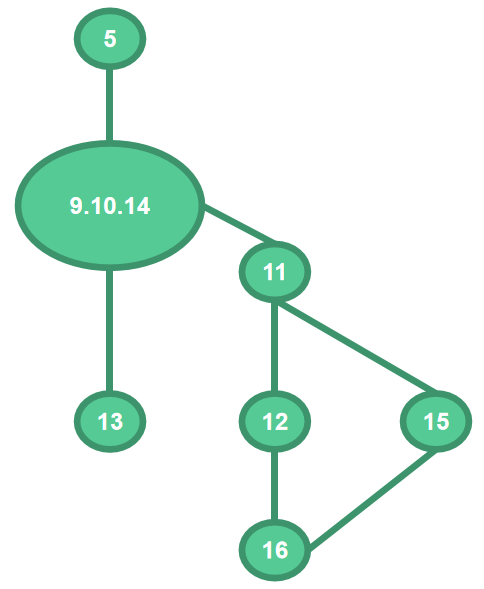


我们看9和14的父亲是不是同一个集合的，一看，都是10，那就是一个集合的，因为初始的并查集里面都是单独元素做一个集合。

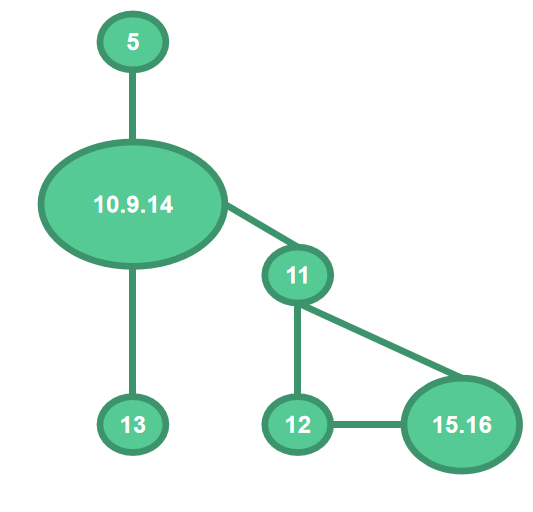
（9，14）现在就做完了，这两个点找到他们的最低公共祖先也就是lca了，我们这里的最低公共祖先是广义的，是并查集里面的一个集合。找到以后就可以把他们合并了，如下图：



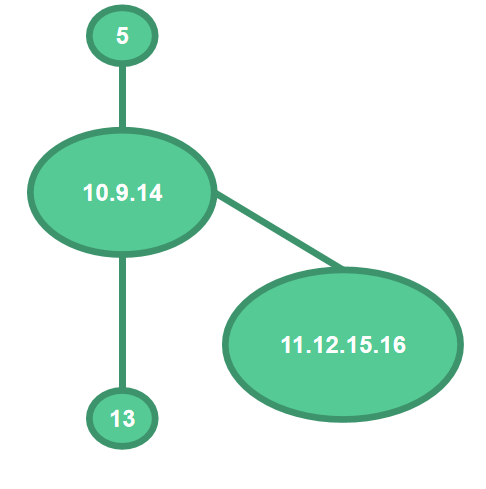
* （15，16）



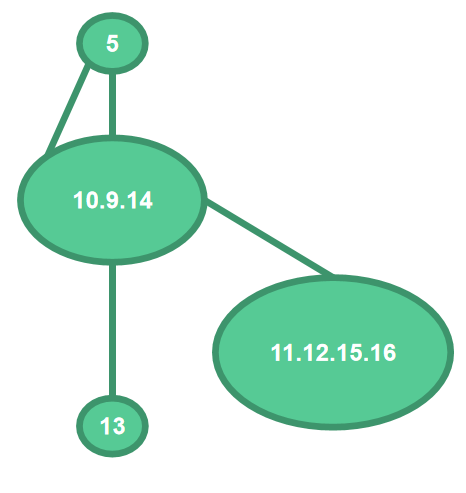
（15，16）他们的父亲不在同一个集合里面，所以我们先把他们合并，因为最后他们总是要合并的。这里为了保证环的特性，我们把深度大的点合并到深度小的，以保持环。这里16的深度比15大，我们把16并到15的集合，然后向上跳一个结点，结果如下图：



然后变成12和（15.16），这时候他们的父亲是属于同一个集合（11）的，所以和前面的情况一样，缩点合并，结果如下：

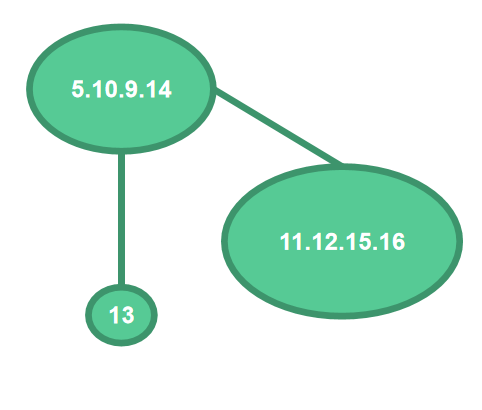


* （5，9）



对于5，这里情况比较特殊，因为5是树根，都没有父亲了，这里我们可以假设它他有一个虚拟父亲。

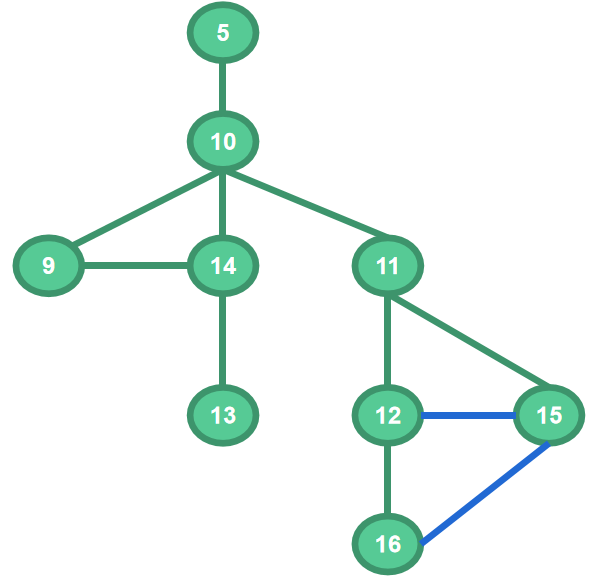
对于9，这个9都是属于10这个集合的，这3个是一块的已经缩成点了，所以实际上父亲就是5。



合并到这里，就会发现他们的父亲都是虚拟父亲，属于同一个集合，所以就缩完了，图中剩下的两条边为（13,14）（11,10）为桥。

**并查集的优化思想：**

并查集用到这里有一个好处就是，像下图里面的情况，蓝色两条边的其中一条如果缩过环，都能大大简化另一条的缩环过程，所以复杂度会低很多。

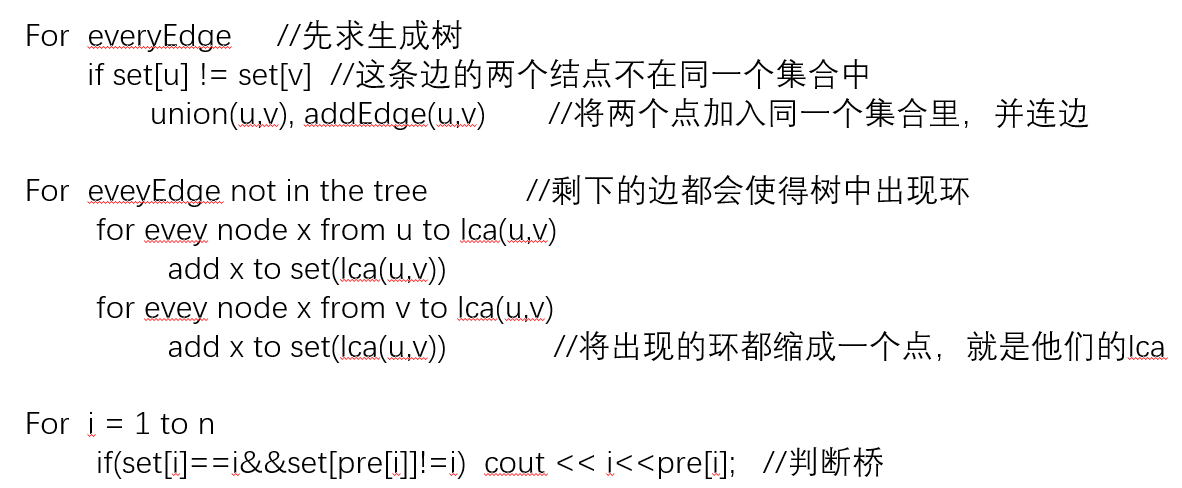


**效率分析**

算法的时间复杂度大概是nlog(n),每次遍历全图n个结点，对于每一个结点的操作都只有find,或者union,而没有再去做别的操作。

所以这个算法的时间会优于基准算法（时间复杂度为O(n\*m)）.

**伪代码**



1. **实验结果**

|  |  |  |
| --- | --- | --- |
| **算法** | **基准算法** | **并查集改良LCA** |
| **桥数量** | **0** | **0** |
| **用时（ms）** | **3.21** | **0** |

结果如上图，数据集中没有桥，基准算法耗时也只有3.21ms，可能是因为数据集太小，看不出明显差别。并查集改良LCA算法耗时0ms，可能是太短了测不出来。

1. **心得体会**

本次实验我成功结合了并查集这种数据结构结合LCA设计出了比基准算法更高效的算法，主要的改良在于每次调用LCA实际上已经将环内的点并在一个集合，于是跳点可以直接跳至集合的代表元素，从而大大提高了查找“LCA”的效率。通过这个实验，我熟悉了并查集这种数据结构，觉得这个东西非常简单而优雅，我们可以用它根据元素不同的特性去将它们分组，基于集合对目标进行操作可以提高效率。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。