实验 3 图

一、 无向图的双连通性算法

连通图和非联通图：如果无向图 G 中任意一对顶点都是连通的，则称此图是连通图；相反，如果一个无向图不是连通图，则称为非连通图（disconnected graph）。

关节点：若在删去顶点 a 以及与之相邻的边之后，图 G 被分割成两个或两个以上的连通分量（连通图），则顶点 a 为连通无向图的关节点。

双连通图：没有关节点的连通图称为双连通图（ Biconnected Graph）。

实践问题：

请设计并实现算法，判断下面的 3 个无向图是否为双连通图。

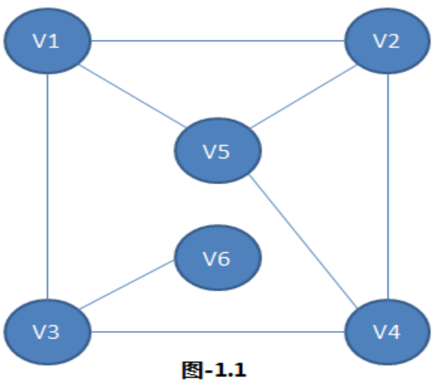

图-1.1

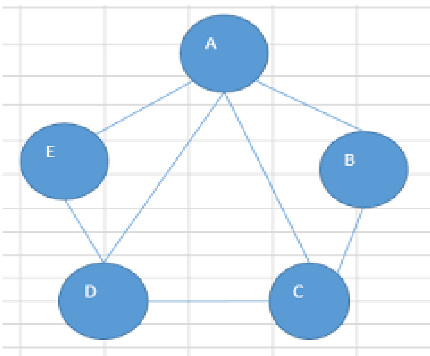

图1.2

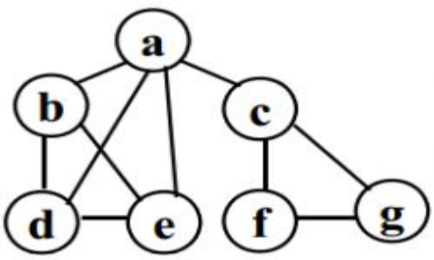

图-1.3

二、 有向图的强连通性算法

强连通（strongly connected）：若 G 是有向图，如果对图 G 中任意两个顶点 u 和 v，既存在从 u 到 v 的路径，也存在从 v 到 u 的路径，则称该有向图为强连通有向图。对于非强连通图，其极大强连通子图称为其强连通分量。

请参考：1. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. Introduction to Algorithrus, The MIT Press, Third Edition. 对应中文译本 P357，22 章第 5 节，强连通分量。

实践问题：

求下面几个有向图的强连通分量的个数及每个强连通分量对应的顶点集合;

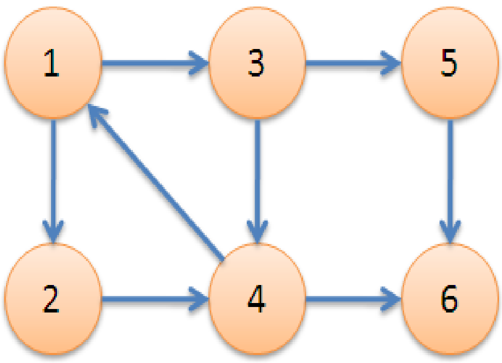

图-2.1

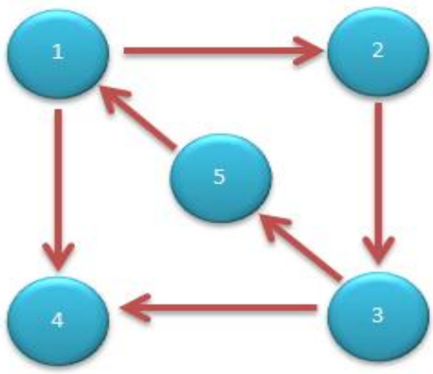

图-2.2

三、　最小生成树算法的实现（Prim、kruskal）

实践问题：

请实现 Prim 和 Kruskal 算法，找出下面几个加权无向图的最小生成树，并输出顶点集合、边集合及最小权值。并思考如果图比较稀疏，该如何优化。

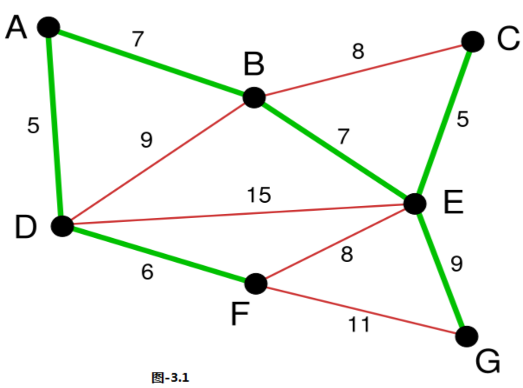

图-3.1

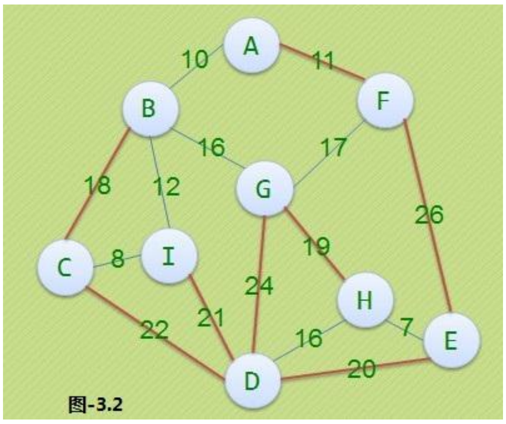

图-3.2

四、　最短路径的算法优化（Dijkstra、Floyd）

实践问题：

请实现 Dijkstra、Floyd 算法，找出下图中点 A 到点 E 的最短路径，输出顶点序列；并思考有负权值该如何计算，无环路有向图最快应该怎样计算。