### 实验内容

每个小实验要求提交运行结果截图和实验项目源码

#### 1.图数据结构

数据集：

数据集目录下包含三个文件(smallgraph.txt, facebook\_combined.txt, roadNet-CA.txt)。以上文件中，图用顺序存放的边表表示，每条边包含两个顶点以及权重，文件的存储格式见下表。

|  |  |  |
| --- | --- | --- |
| 注释 | | |
| FromNodeID | ToNodeID | Weight |
| FromNodeID | ToNodeID | Weight |
| …… | …… | …… |
| [EOF] | | |

其中，smallgraph.txt为有向小型图，facebook\_combined.txt为无向中型图，roadNet-CA.txt存放了有向大型图。

步骤说明：

编写c++程序将上述三张图的边表结构读取并转化为图的邻接矩阵或者邻接表形式（每张图两种数据结构任选其一但两者至少用一次）

#### 2.深度优先遍历DFS和广度优先遍历BFS

步骤说明：

在实验1的基础上编写程序对facebook\_combined和roadNet-CA两张图进行DFS和BFS（每张图任选一种遍历算法但两次使用的算法须不同）；

输出两张图的连通分量数目（打印到屏幕上）

例” facebook\_combined: number of connected components = x”。

#### 3.Prim\Kruskal 算法

步骤说明：

在实验1的基础上编写程序实现Prim或Kruskal算法求出任意两张图的最小生成树；

创建MST数组来存放最小生成树的边，输出最小生成树的权值之和（打印到屏幕上）

例“facebook\_combined: sum\_weight of MST = x”。

#### 4.Dijkrstra 算法

步骤说明：

在实验1的基础上编写程序实现Dijkrstra算法；

在任意两张图上输入源点ID求出其单源最短路径，结果保存在数组distance[n]中；

打印任意两点最短路径结果到屏幕上。

例“Distance from node 0 to node 9 = x”