**《数据结构》实验报告（第6章）**

**学号：\_09020328\_\_\_\_ 姓名 ：\_ 王亮\_\_\_\_**

**实验题号： P352: 3,5; P359: 3\_ 实验日期：\_2021.11.19\_\_\_**

**实验一**

**1．问题描述：**

为邻接表表示的图写一个深度优先搜索函数，并测试此函数。

为邻接表表示的图写一个广度优先搜索函数，并测试此函数。

**2．算法思想：**

DFS：提供一个与顶点数v相同大小的数组，用于记录顶点v是否被访问过。选取一未被访问的顶点v作为入口，访问其一尚未被访问相邻节点，继续访问该节点一尚未被访问相邻节点，直至所有节点被访问。

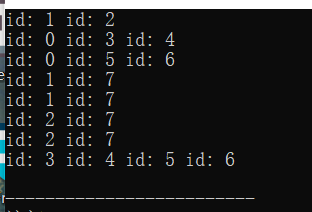
BFS：数组同上，每次让当前未被访问节点出队，标记为已访问，将所有相邻的节点加入队列中，以此循环至所有节点被访问。

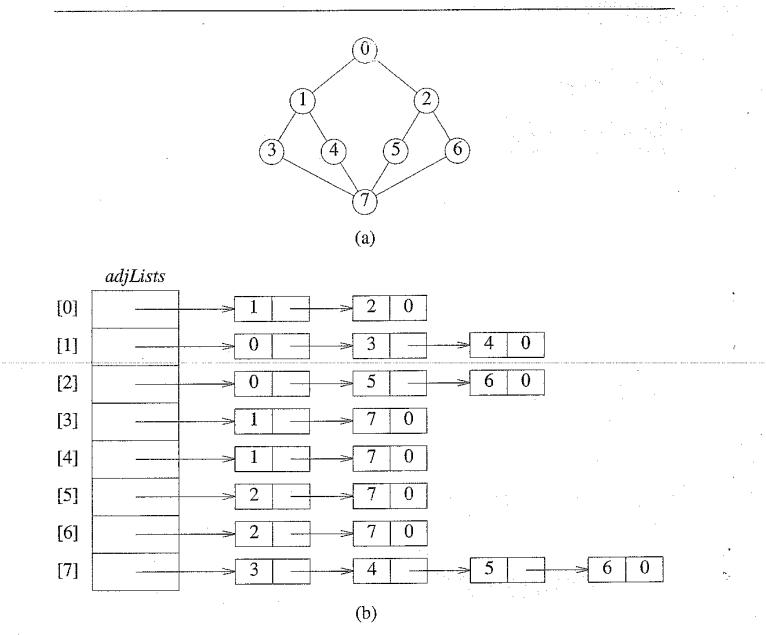
**3．功能函数：**

void dfs(int v)

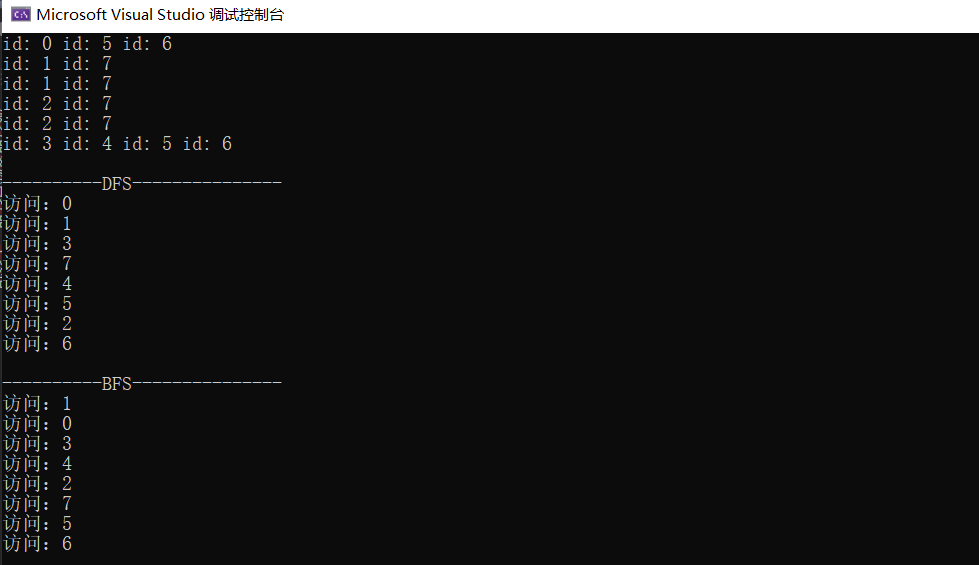
void bfs(int v)

**4．测试数据：**





**5．测试情况：**



**6．实验总结：**

**时间复杂度：采用邻接表表示，访问邻接表中每个顶点，访问邻接表中每个顶点子表，总计均为O(V+E)，v为顶点数，e为边数**

**7. 源代码：**

实验源程序清单

(1) As-LabEXP6.cpp

**实验二**

**1．问题描述：**

改进课本Program6.7，将其变为一个能以O(n^2)时间复杂度代价找到最小生成树的函数，并且证明这一点。

**2．算法思想：**

准备一下数组：

isJoin,lowCost,ori,MST

其大小均为v，即顶点个数。

MST：生成树数组

lowCost：当前已形成的子图到其他所有顶点的最小代价

ori：调试使用，用来记录连入子图的顶点连接点

isJoin：是否加入子图

对于初始顶点，初始化isJoin与lowCost，ori数组

将初始顶点v加入MST

然后循环：

访问所有剩余节点

寻找未加入子图的最小代价顶点min

将min加入子图

以min到其他顶点代价更新子图最小代价

**3．功能函数：**

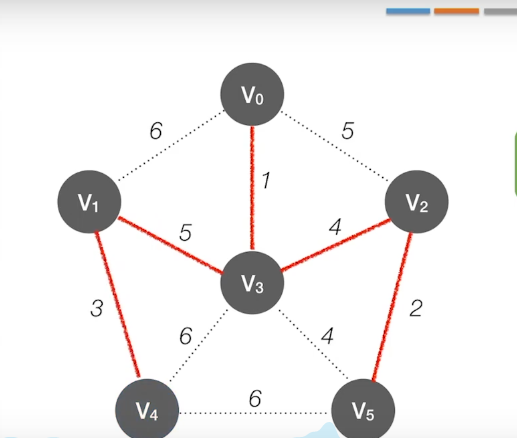
pair<int\*,int\*> prim(int\*\* matrix,const int v)

prim算法求最小生成树

void printMST(pair<int\*,int\*>tar)

打印最小生成树

**4．测试数据：**





**5．测试情况：**



**6．实验总结：**

时间复杂度为O(n^2),对于每一个节点，都访问它与其他所有节点的最短距离进行更新。

辅助的数组均为O(n)，即顶点数。

**7. 源代码：**

实验源程序清单

(1) MST.cpp