

课前签到

数据结构与算法实验课2022秋季



微信扫一扫，使用小程序

1. 微信扫码+**实名**
2. 点击今日签到

签到时间：

16:20~16:35

签到地方：

**珠海校区-教学大
楼-A310**

人脸识别；智能定位

实验十四 图

- **实验目的**
 - **掌握图的逻辑结构和存储结构；**
 - **掌握图的两种遍历算法；**
 - **掌握如何应用图解决各种实际问题。**



实验十四 图

- **实验要求**

- **图算法的实现以库文件方式实现，不得在测试主程序中直接实现；**
- **程序有较好可读性，各运算和变量的命名直观易懂，符合关键工程要求；**
- **程序有适当的注释。**

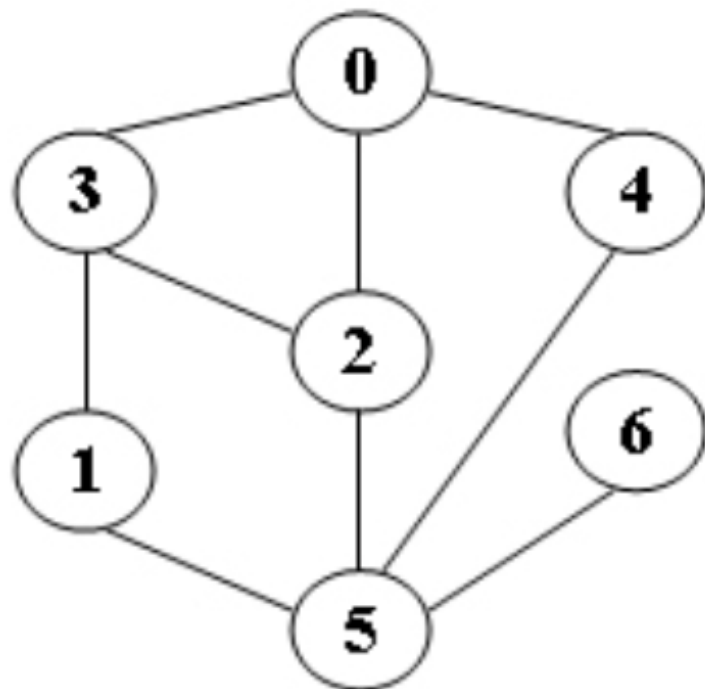


实验十四 图

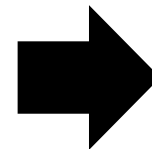
- 实验任务1：
 - 完成如下函数，实现邻接矩阵存储图的深度优先遍历

```
void DFS( MGraph Graph, Vertex V, void (*Visit)(Vertex) )
```

示例：



Input:
2



Output:
DFS from 5: 5 1 3 0 2 4 6

实验十四 图

● 实验任务1：

```
void DFS( MGraph Graph, Vertex V, void (*Visit)(Vertex) )
```

**Mgraph是邻接
矩阵存储的图，
定义如右**

```
typedef struct GNode *PtrToGNode;  
struct GNode{  
    int Nv; /* 顶点数 */  
    int Ne; /* 边数 */  
    WeightType G[MaxVertexNum][MaxVertexNum];  
    /* 邻接矩阵 */  
};  
typedef PtrToGNode MGraph; /* 以邻接矩阵存储的  
图类型 */
```

函数DFS应从第V个顶点出发递归地深度优先遍历图Graph，遍历时用裁判定义的函数Visit访问每个顶点。当访问邻接点时，要求按序号递增的顺序。题目保证V是图中的合法顶点



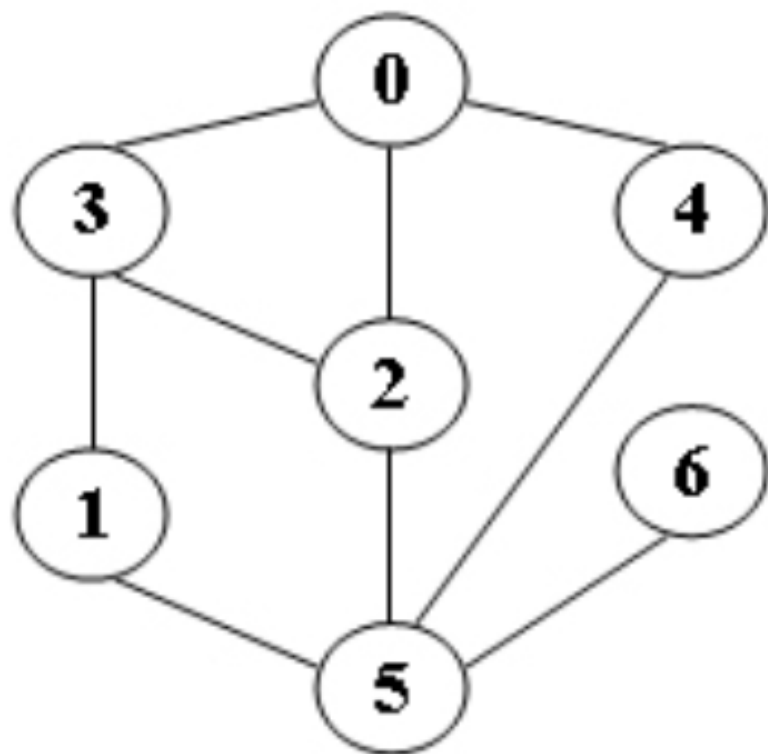
实验十四 图

- 实验任务2：

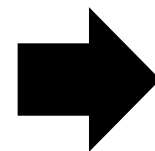
- 完成如下函数，实现邻接表存储图的广度优先遍历

```
void BFS( LGraph Graph, Vertex S, void (*Visit)(Vertex) )
```

示例：



Input:
2



Output:
BFS from 2: 2 0 3 5 4 1 6

实验十四 图

● 实验任务2：

void **BFS**(LGraph Graph, Vertex S, void (*Visit)(Vertex))

LGraph是邻接矩阵存储的图，定义如右

```
/* 邻接点的定义 */
typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;          /* 邻接点下标 */
    PtrToAdjVNode Next; /* 指向下一个邻接点的指针 */
};

/* 顶点表头结点的定义 */
typedef struct Vnode{
    PtrToAdjVNode FirstEdge; /* 边表头指针 */
} AdjList[MaxVertexNum]; /* AdjList是邻接表类型 */

/* 图结点的定义 */
typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;      /* 顶点数 */
    int Ne;      /* 边数 */
    AdjList G;   /* 邻接表 */
};
typedef PtrToGNode LGraph; /* 以邻接表方式存储的图类型 */
```

函数BFS应从第S个顶点出发对邻接表存储的图Graph进行广度优先搜索，遍历时用裁判定义的函数Visit访问每个顶点。当访问邻接点时，要求按邻接表顺序访问。题目保证S是图中的合法顶点

实验十四 图

- **实验任务3：**

- **给定一个有 N 个顶点和 E 条边的无向图，请用DFS和BFS分别列出其所有的连通集。假设顶点从0到 $N-1$ 编号。进行搜索时，假设我们总是从编号最小的顶点出发，按编号递增的顺序访问邻接点**

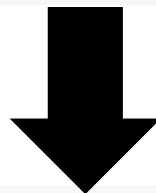


实验十四 图

● 实验任务3：

示例：

```
8 6
0 7
0 1
2 0
4 1
2 4
3 5
```



```
{ 0 1 4 2 7 }
{ 3 5 }
{ 6 }
{ 0 1 2 7 4 }
{ 3 5 }
{ 6 }
```

注意：

- 输入：第1行给出2个整数 $N(0 < N \leq 10)$ 和 E ，分别是图的顶点数和边数。随后 E 行，每行给出一条边的两个端点。每行中的数字之间用1空格分隔
- 输出：按照" $\{ v_1 v_2 \dots v_k \}$ "的格式，每行输出一个连通集。先输出DFS的结果，再输出BFS的结果

实验十四 图

- **实验任务4：**
 - **假设有一个地下通道迷宫，它的通道都是直的，而通道所有交叉点（包括通道的端点）上都有一盏灯和一个开关。请问你如何从某个起点开始在迷宫中点亮所有的灯并回到起点**



实验十四 图

● 实验任务4：

示例：

6 8 1

1 2

2 3

3 4

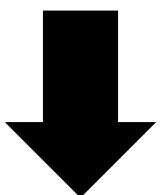
4 5

5 6

6 4

3 6

1 5



1 2 3 4 5 6 5 4 3 2 1

注意：

— 输入：第一行三个正整数，分别表示地下迷宫的节点数 N ($1 < N \leq 1000$ ，表示通道所有交叉点和端点)、边数 M (≤ 3000 ，表示通道数)和探索起始节点编号 S (节点从1到 N 编号)。随后的 M 行对应 M 条边(通道)，每行给出一对正整数，分别是该条边直接连通的两个节点的编号

— 输出：若可以点亮所有节点的灯，则输出从 S 开始并以 S 结束的包含所有节点的序列，序列中相邻的节点一定有边(通道)；否则虽不能点亮所有节点的灯，但还是输出点亮部分灯的节点序列，最后输出0，此时表示迷宫不是连通图。由于深度优先遍历的节点序列不唯一，为了使得输出具有唯一结果，约定以节点小编号优先的次序访问(点灯)。在点亮所有可以点亮的灯后，以原路返回的方式回到起点