



# 数据结构与算法实验

图

彭振辉

中山大学人工智能学院

2023年秋季学期

# 实验目的与要求

---



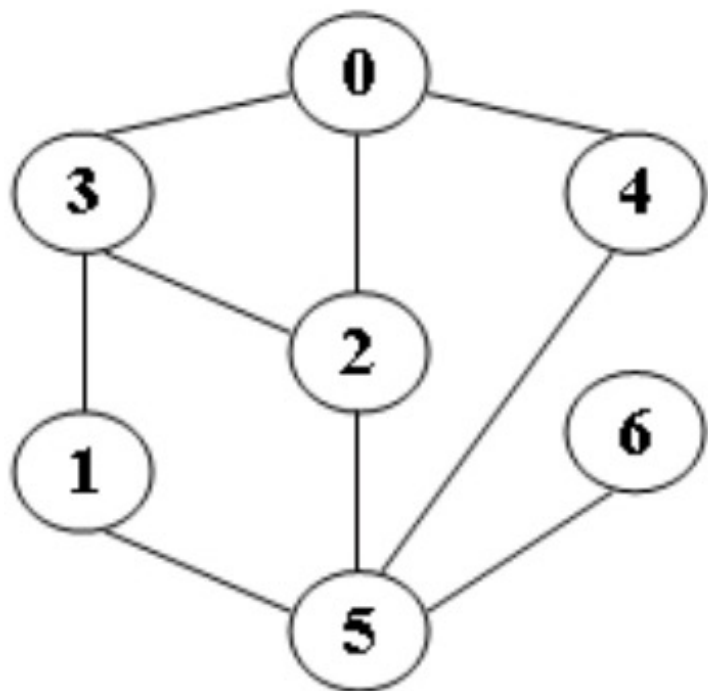
- 掌握**图的逻辑结构和存储结构**；
  - 掌握**图的两种遍历算法**；
  - 掌握如何应用图解决各种实际问题。
- 
- 图算法的实现以库文件方式实现，不得在测试主程序中直接实现；
  - 程序有较好可读性，各运算和变量的命名直观易懂，符合关键工程要求；
  - 程序有适当的注释。

# 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

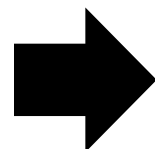
```
void DFS( MGraph Graph, Vertex V, void (*Visit)(Vertex) )
```

示例：



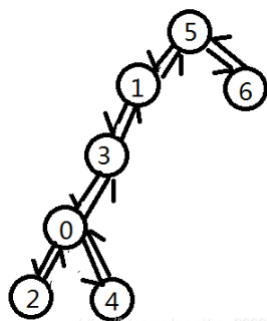
Input:

5



Output:

DFS from 5: 5 1 3 0 2 4 6



- 1、从5开始搜索，按从小到大的顺序搜索：首先搜索到1，标记，然后递归从1搜索；
- 2、节点1遍历到3，3被标记，开始搜索，然后依次访问到0和2，标记；
- 3、与2相连的还有3和5，但是3和5已经被标记过了，回溯到0，继续访问4，标记；
- 4、4节点和5相连，但是5已经被标记过了，不再访问，依次回溯到0->3->1->5节点；
- 5、与5相连的还有2和4，2和4已被访问，于是访问6，标记；
- 6、6没有与其他节点相连，回溯至5，遍历结束。



# 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 */
```

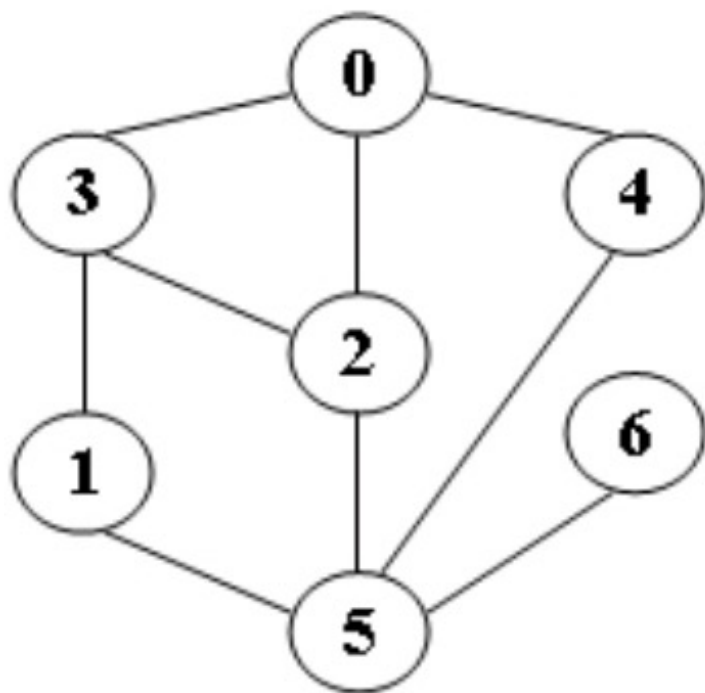
函数DFS应从**第V个顶点**出发递归地深度优先遍历图**Graph**，遍历时用裁判定义的函数Visit访问每个顶点。当访问邻接点时，要求按序号递增的顺序。题目保证V是图中的合法顶点

## 任务2

- 完成如下函数，实现**邻接表**存储图的**广度优先遍历**

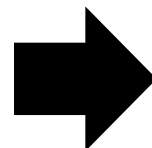
```
void BFS( LGraph Graph, Vertex S, void (*Visit)(Vertex) )
```

示例：



Input:

2



Output:

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

或：**BFS from 2: 2 5 3 0 6 4 1**

或：**... 取决于插入的边的顺序**



## 任务2

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

**LGraph**是邻接表存储的

图，定义如右

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**

```
/* 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 */
```



## 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

示例输入：

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

— 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

示例输出：

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

— 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

## 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**

示例输入：

6 8 1  
1 2  
2 3  
3 4  
4 5  
5 6  
6 4  
3 6  
1 5

- 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** $M$  ( $\leq 3000$ ，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**

示例输出：

1 2 3 4 5 6 5 4 3 2 1