



石家莊鐵道大學  
SHIJIAZHUANG TIEDAO UNIVERSITY

在线开放课程

数据结构— 图

图的遍历—深度优先搜索

主讲：石玉晶

# 目录



在线开放课程

- ◆ 1、图的遍历的概念
- ◆ 2、深度优先搜索的过程
- ◆ 3、深度优先搜索的算法实现
- ◆ 4、深度优先搜索算法分析

# 一、图的遍历的概念

## • 图的遍历

- 从图中某一顶点出发，访问图中其余顶点，使每个顶点被访问一次且只被访问一次。
- 可以从图中任意一个顶点出发进行遍历。
- 遍历中需解决的问题
  - 确定一搜索路径；
  - 确保每个顶点被访问到；
  - 确保每个顶点只能被访问一次。

# 一、图的遍历的概念

## • 图的遍历

### – 解决方法

- 深度优先搜索 (Depth First Search, DFS)
- 广度优先搜索 (Breadth First Search, BFS)
- 设**辅助数组visited**，初始时，数组元素的值均为0或false，表示未被遍历，一旦遍历，就置为1或true。
- 当图中有回路，又回到被访问过的顶点时，则通过**visited**，**避免一个顶点被访问多次**。

## 二、深度优先搜索的过程

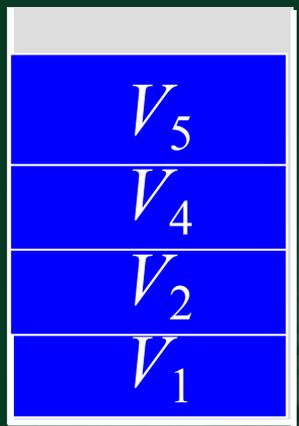
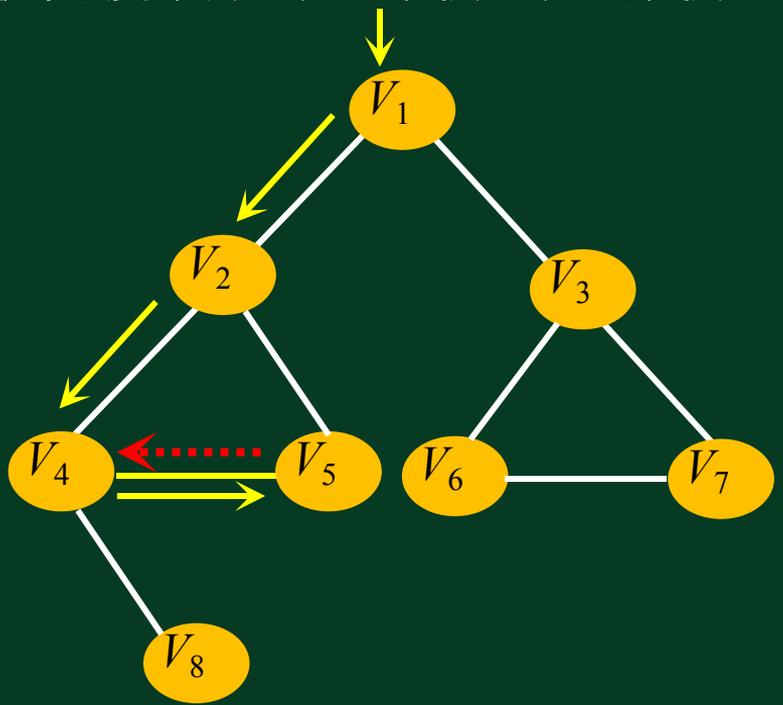
### • 方法

- (1) 从图的某一顶点 $V_0$ 出发，访问此顶点；
- (2) 找出刚访问过的顶点的第一个未被访问的邻接点，访问该顶点。以该顶点为新顶点，重复此步骤，直到刚访问过的顶点没有未被访问的邻接点为止；
- (3) 返回前一个访问过的且仍有未被访问的邻接点的顶点，找到该顶点的下一个未被访问的邻接点，访问该顶点。
- (4) 重复(2)、(3)，直到联通的子图中的节点都访问到。
- 若图为**非连通图**，若此时图中尚有顶点未被访问，则另选图中一个未被访问的顶点作起点，重复上述过程，直至图中所有顶点都被访问为止。

# 二、深度优先搜索的过程

深度优先搜索序列? 入栈序列? 出栈序列?

深一层递归  
————→  
递归返回  
·····→



遍历序列:  $V_1$   $V_2$   $V_4$   $V_5$

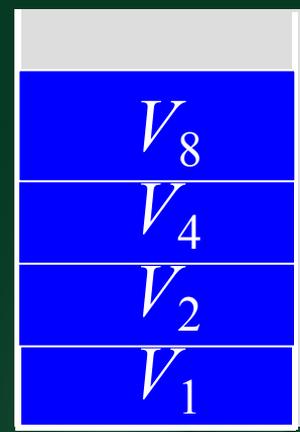
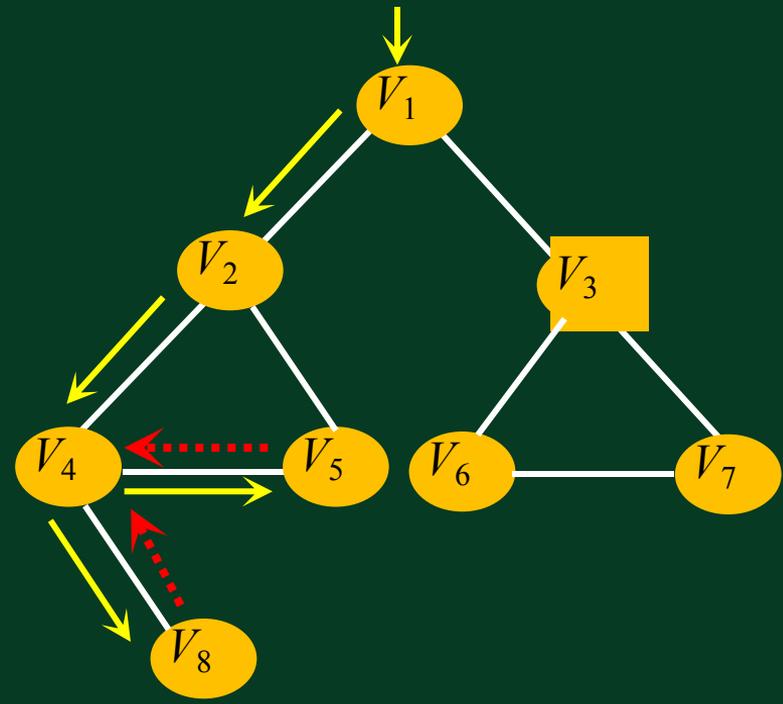
# 二、深度优先搜索的过程

深度优先搜索序列?入栈序列?出栈序列?

深一层递归



递归返回



遍历序列:  $V_1$   $V_2$   $V_4$   $V_5$   $V_8$

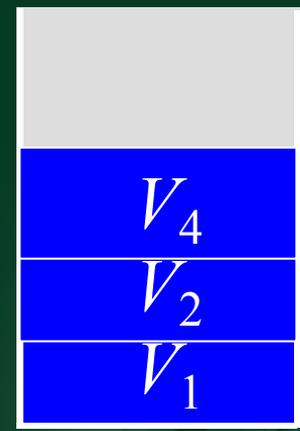
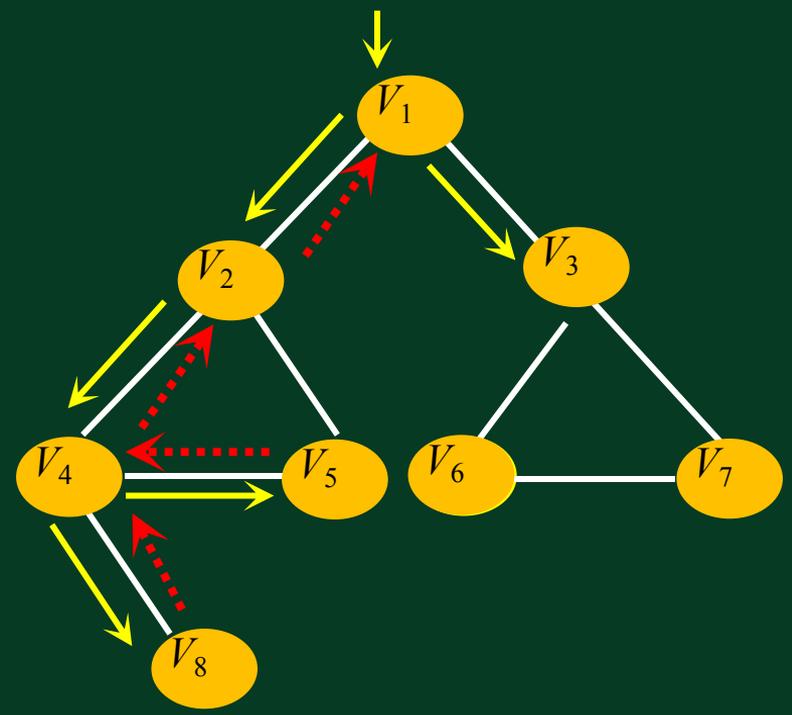
# 二、深度优先搜索的过程

深度优先搜索序列?入栈序列?出栈序列?

深一层递归



递归返回

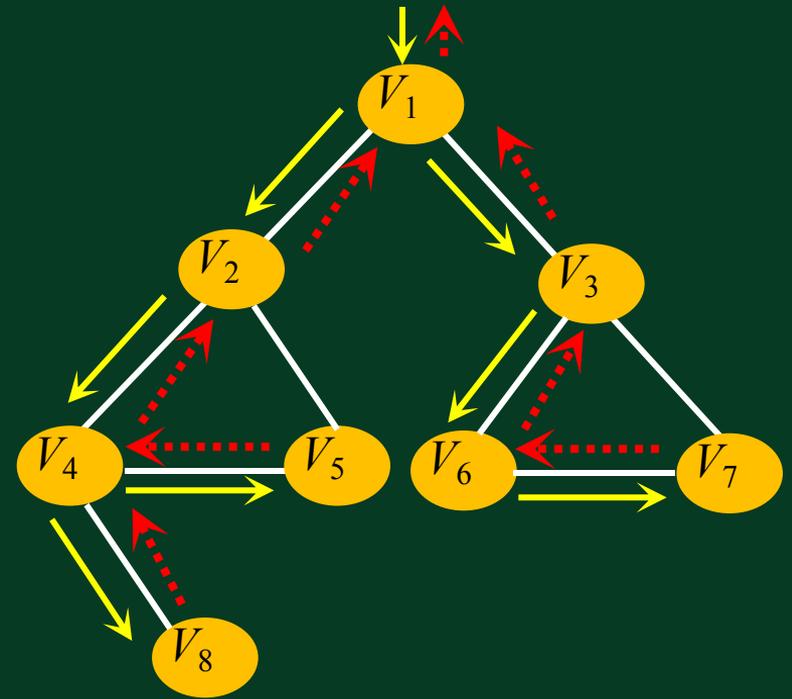


遍历序列:  $V_1 V_2 V_4 V_5 V_8$

# 二、深度优先搜索的过程

深度优先搜索序列?入栈序列?出栈序列?

深一层递归  
—————→  
递归返回  
-----→



$V_7$
$V_6$
$V_3$
$V_1$

遍历序列:  $V_1 V_2 V_4 V_5 V_8 V_3 V_6 V_7$

# 三、深度优先搜索的算法实现

- 【算法描述】

- (1) 从图中某个顶点 $V$ 出发，访问 $V$ ，并设置  $visited[v]=true$ ;
- (2) 一次检查 $V$ 的所有邻接点 $W$ ，如果  $visited[W]=false$ ，再从 $W$ 出发进行递归遍历，直到图中所有顶点都被访问过。

# 三、深度优先搜索的算法实现



在线开放课程

```
Boolean visited[MAX];  
void DFS (Graph G, int v) {  
    cout<<v;  visited[v]=true;  
    for ( w=FirstAdjVex(G, v);w>=0;w=NextAdjVex(G, v, w))  
        if (!visited[w])  
            DFS(G, w);  
}
```

## 四、深度优先搜索算法分析

- 遍历图的过程，实质上是对每个顶点查找其邻接点的过程，耗费的时间取决于所采用的存储结构。
  - 用邻接矩阵来表示图，遍历图中每一个顶点都要从头扫描该顶点所在行，时间复杂度为 $O(n^2)$ 。
  - 用邻接表来表示图，查找邻接点的复杂度为 $O(e)$ ，加上访问  $n$  个头结点的时间，时间复杂度为 $O(n+e)$ 。

## 四、深度优先搜索算法分析

- 遍历图的过程，实质上是对每个顶点查找其邻接点的过程，耗费的时间取决于所采用的存储结构。

**结论：**

**稠密图**适于在邻接矩阵上进行深度遍历；

**稀疏图**适于在邻接表上进行深度遍历。

# 小结



在线开放课程

- 图的遍历——深度优先搜索
- 深度优先搜索的算法

# 小结



在线开放课程

- 接下来学习：
- 图的遍历—广度优先搜索

# 谢谢！