

The background is a solid blue color with a subtle pattern of small, light blue geometric shapes (cubes and rectangles) scattered across it. A faint, repeating watermark of the text "中国大学MOOC" is visible diagonally across the entire image. In the center, the large white Chinese character "图" is displayed.

图

大连理工大学

刘馨月

主要内容

- 图的基本概念
- 图的存储
- 图的遍历
- 最小生成树
- 最短路径
- 关键路径

图的遍历

- 图的遍历

给出一个图G和其中任意一个顶点 V_0 ，从 V_0 出发系统地访问G中所有的顶点，每个顶点访问一次，这叫做图的遍历。

- 深度优先搜索

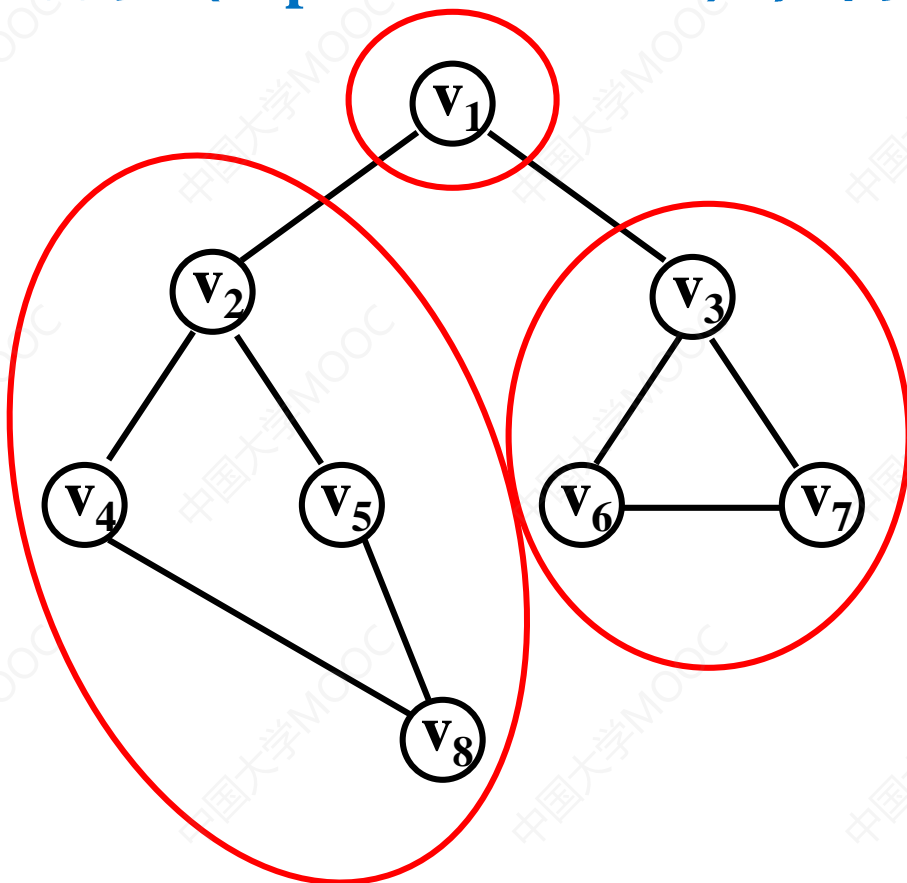
- 广度优先搜索

深度优先搜索 (depth-first search) , 简称DFS

基本思想

- 访问一个顶点 V ，然后访问该顶点邻接到的未被访问过的顶点 V' ，
- 再从 V' 出发递归地按照深度优先的方式遍历；
- 当遇到一个所有邻接于它的顶点都被访问过了的顶点 U 时，则回到已访问顶点序列中最后一个拥有未被访问的相邻顶点的顶点 W ；
- 再从 W 出发递归地按照深度优先的方式遍历；
- 最后，当任何已被访问过的顶点都没有未被访问的相邻顶点时，则遍历结束。

深度优先搜索 (depth-first search), 简称DFS



图可分为三部分:

基结点

第一个邻接结点导出的子图

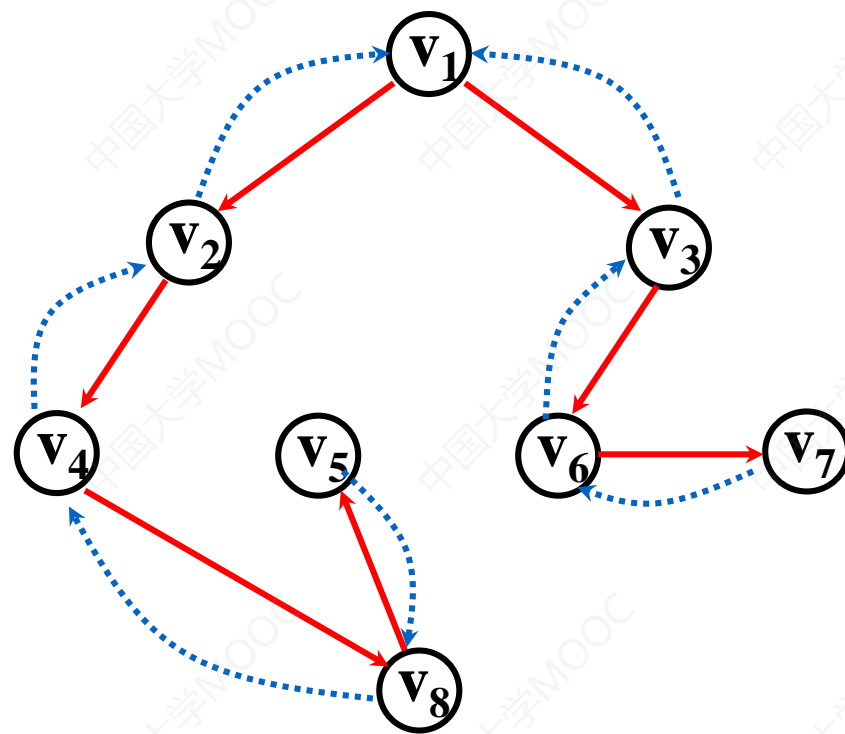
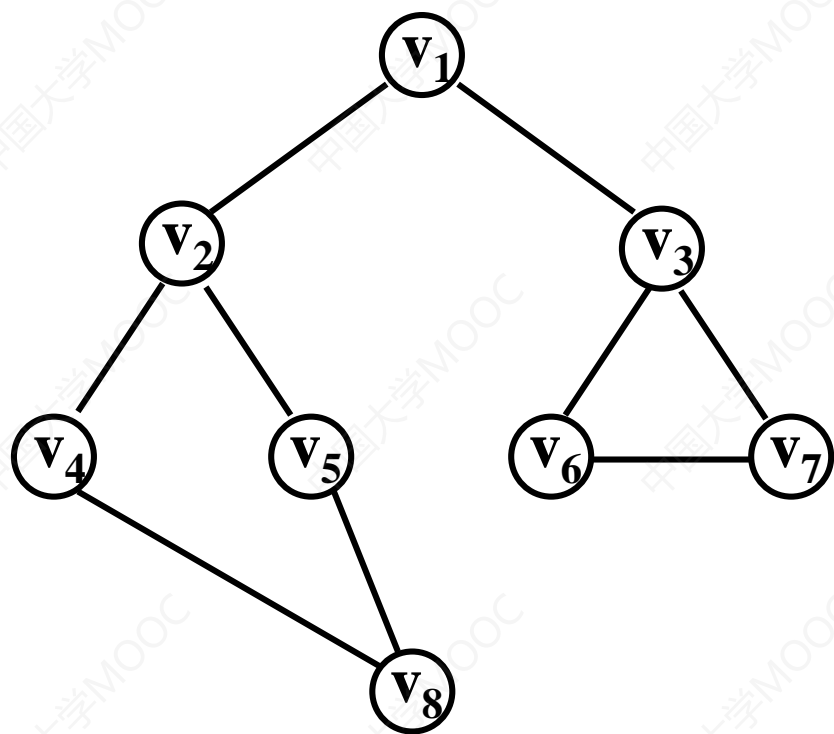
其它邻接顶点导出的子图

深度优先搜索是类似于树的一种先序遍历

深度优先搜索顺序: v_1 v_2 v_4 v_8 v_5 v_3 v_6 v_7

图的遍历

深度优先搜索 (depth-first search) , 简称DFS



深度优先搜索顺序:

v_1 v_2 v_4 v_8 v_5 v_3 v_6 v_7

图的遍历

深度优先搜索 (DFS) 算法

```
void DFS(Graph& G, int v) { // 深度优先搜索的递归算法  
    G.Mark[v] = VISITED; // 将标记位设置为VISITED  
    Visit(G,v); // 访问顶点v  
    for (Edge e = G.FirstEdge(v); G.IsEdge(e); e = G.NextEdge(e))  
        if (G.Mark[G.ToVertex(e)] == UNVISITED)  
            DFS(G, G.ToVertex(e));  
}
```


深度优先搜索 (depth-first search) , 简称DFS

深度优先搜索时间复杂度

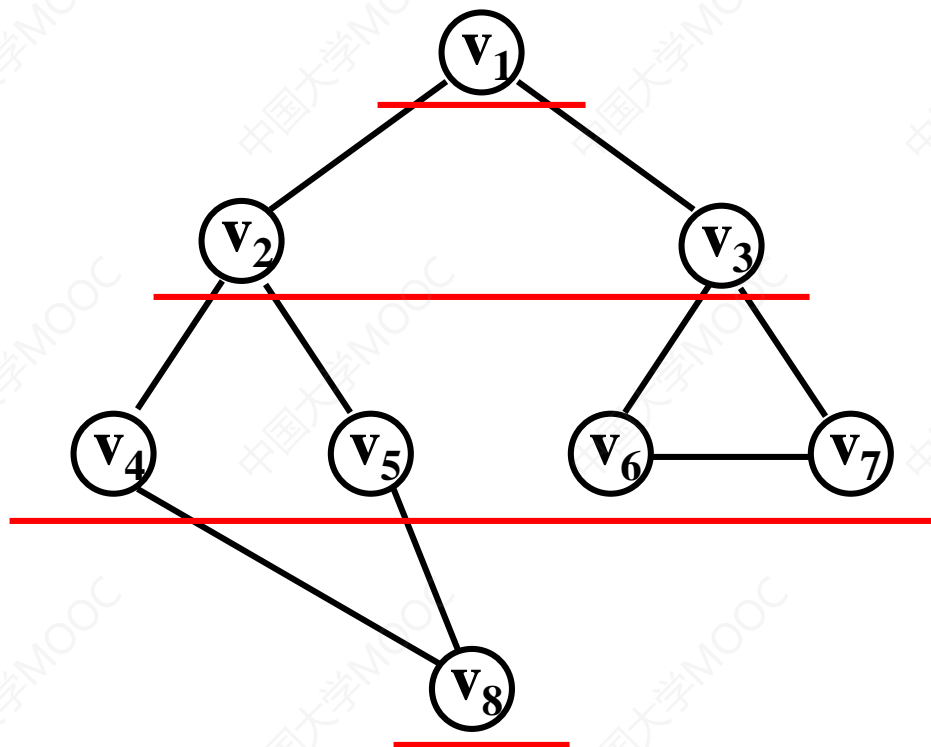
- 对于具有 n 个顶点和 e 条边的无向图或有向图, 深度优先搜索算法对图中每个顶点至多调用一次DFS函数;
- 用邻接矩阵表示图时, 共需检查 n^2 个矩阵元素, 所需时间为 $O(n^2)$;
- 用邻接表表示图时, 找邻接点需将邻接表中所有边结点检查一遍, 需要时间 $O(e)$, 对应的深度优先搜索算法的时间复杂度为 $O(n+e)$ 。

广度优先搜索 (breadth-first search) , 简称BFS

基本思想

- 访问顶点 V_0 ;
- 然后访问 V_0 邻接到的所有未被访问过的顶点 $V_{01}, V_{02}, \dots, V_{0i}$;
- 再依次访问 $V_{01}, V_{02}, \dots, V_{0i}$ 邻接到的所有未被访问的顶点;
- 如此进行下去, 直到访问遍所有的顶点。

广度优先搜索 (breadth-first search) , 简称BFS

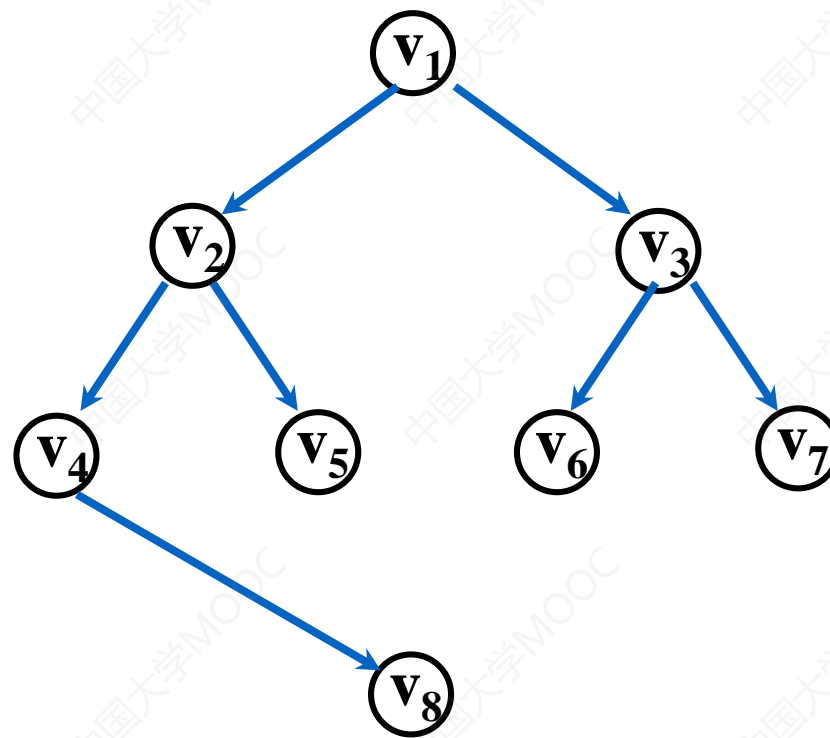
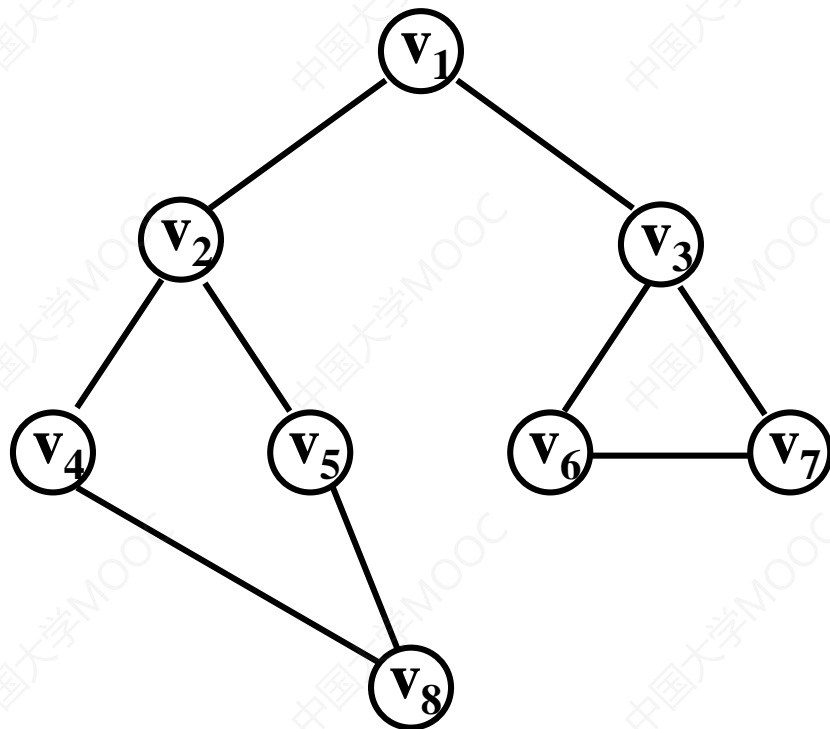


广度优先搜索类似于树的层次遍历

广度优先搜索顺序: **v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8**

图的遍历

广度优先搜索 (breadth-first search) , 简称BFS



广度优先搜索顺序: v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8

图的遍历

广度优先搜索 (BFS) 算法

```
void BFS(Graph& G, int V){  
    using std::queue;  
    queue<int> Q; //初始化广度优先搜索要用到的队列  
    G.Mark[V]= VISITED; //将标记位设置为VISITED  
    Visit(G, V); //访问顶点v  
    Q.push(V); //将顶点v入队列  
    while(!Q.empty()) { //如果队列仍然有元素  
        int V=Q.front(); //队首元素  
        Q.pop(); //出队列  
        //将与该点相邻的每一个未被访问顶点都入队列  
        for(Edge e=G.FirstEdge(V);G.IsEdge(e);e=G.NextEdge(e)){  
            if(G.Mark[G.ToVertex(e)]== UNVISITED) {  
                G.Mark[G.ToVertex(e)]=VISITED;  
                Visit(G, G.ToVertex(e));  
                Q.push(G.ToVertex(e)); //入队列  
            }  
        }  
    }  
}
```

广度优先搜索 (breadth-first search) , 简称BFS

广度优先搜索时间复杂度

- 对于具有 n 个顶点和 e 条边的无向图或有向图, 广度优先搜索算法对图中每个顶点至多调用一次BFS函数;
- 用邻接矩阵表示图时, 共需检查 n^2 个矩阵元素, 所需时间为 $O(n^2)$;
- 用邻接表表示图时, 找邻接点需将邻接表中所有边结点检查一遍, 需要时间 $O(e)$, 对应的广度优先搜索算法的时间复杂度为 $O(n+e)$ 。

The background is a solid blue color with a subtle pattern of small, light blue geometric shapes (cubes and spheres) scattered across it. A faint, repeating watermark of the text "中国大学MOOC" is visible diagonally across the entire image. In the lower-left corner, there is a cluster of larger, 3D blue cubes of various sizes, some of which are slightly offset from the main plane, creating a sense of depth.

图

大连理工大学

刘馨月