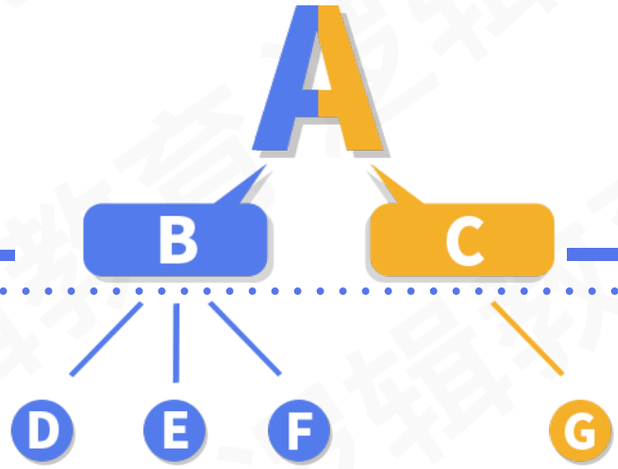
逻辑教育
Logic education

Hello 数据结构与算法



数据结构与算法一图

数据结构与算法主题

@CC老师

全力以赴．非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

经典数据结构——图

课程研发:CC老师
课程授课:CC老师

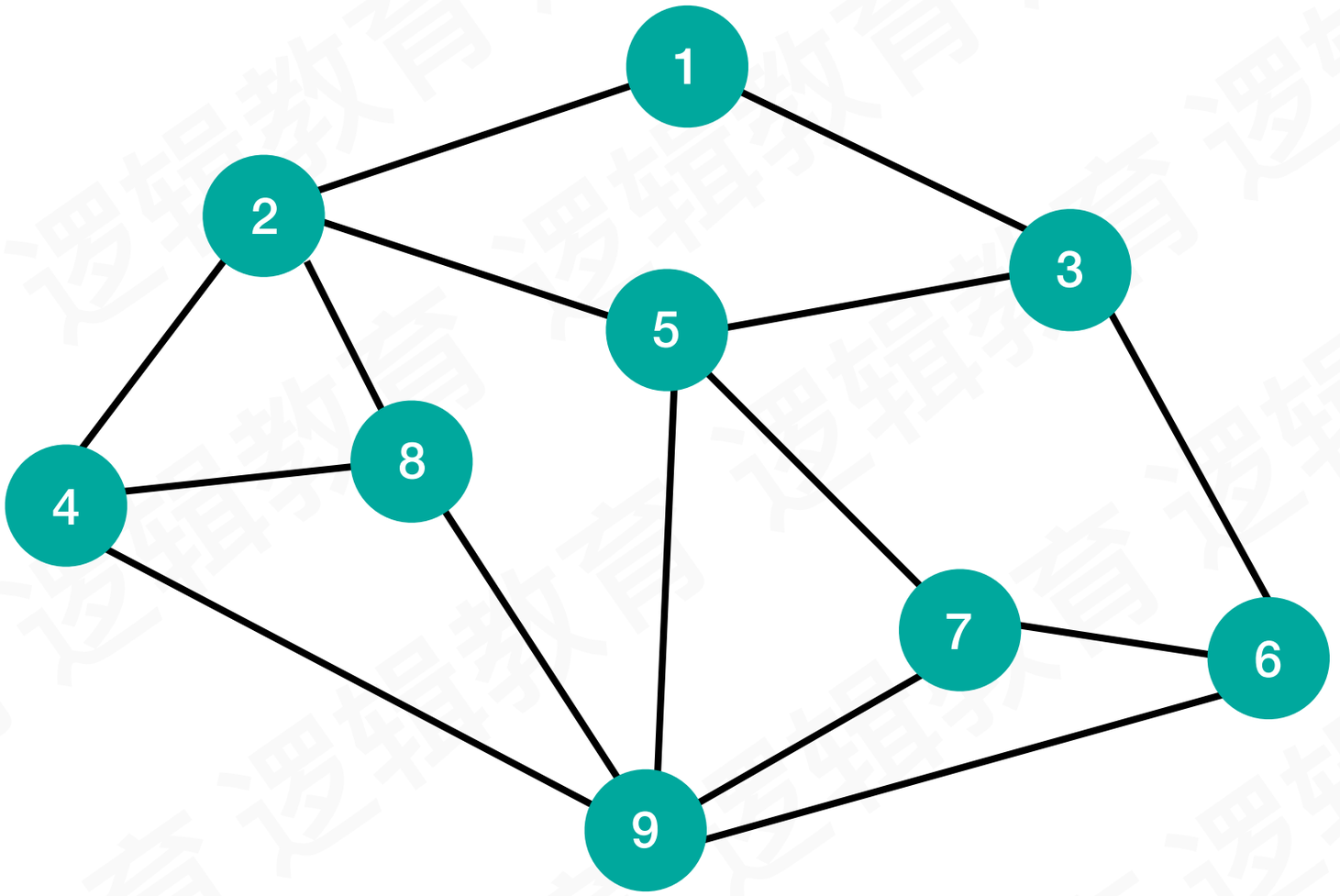
逻辑教育
Logic education

开启”图”版块的学习之旅

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

认识图



图(Graph)是由顶点的有穷非空集合和顶点之间边的集合组成. 通常表示为: G(V,E). 其中,G表示一个图, V是图G中的顶点集合,E是图G中边的集合.

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

认识图

绝世美女图

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

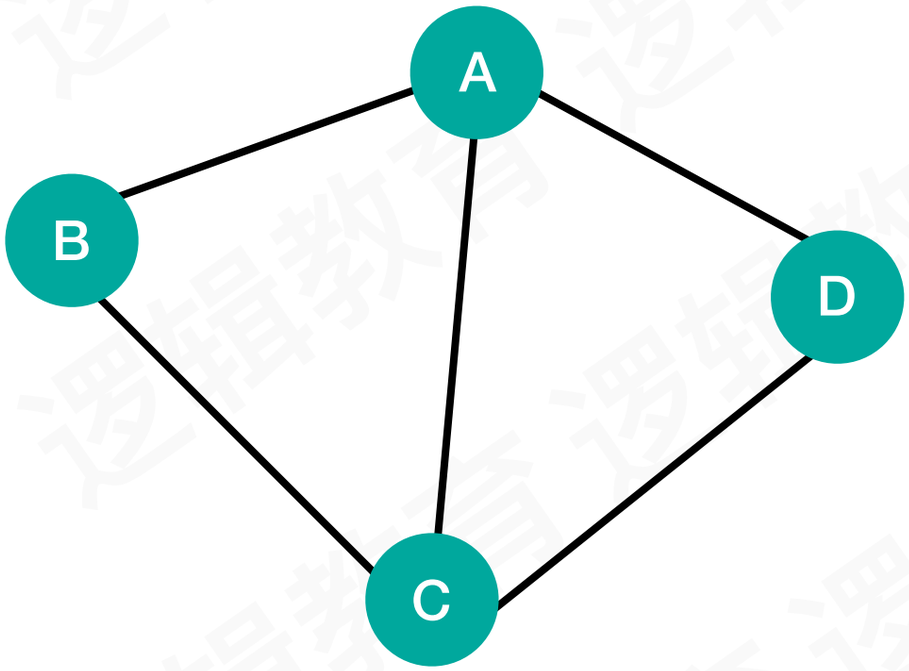
认识图

绝世美女图

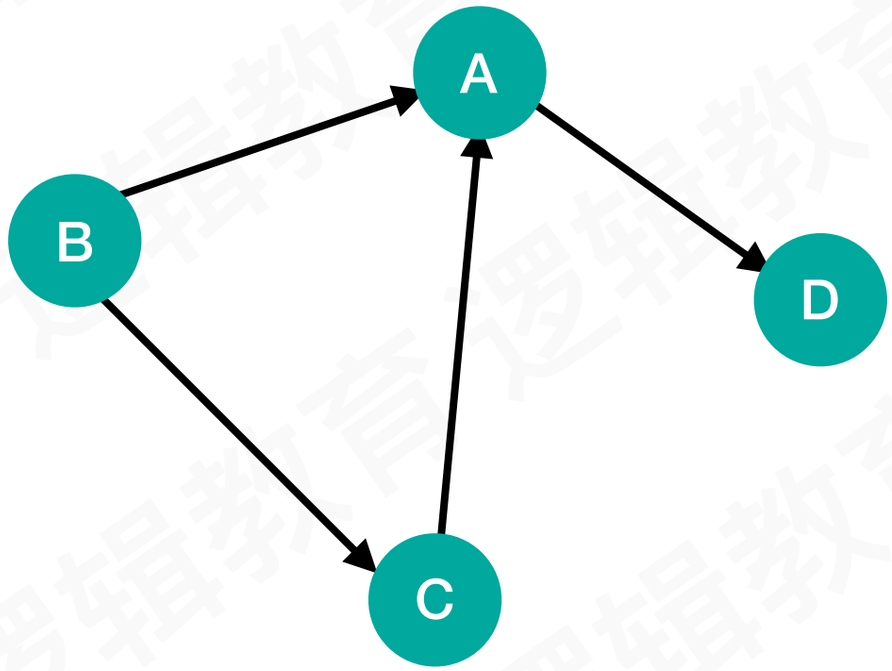
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

各种图的定义(01)



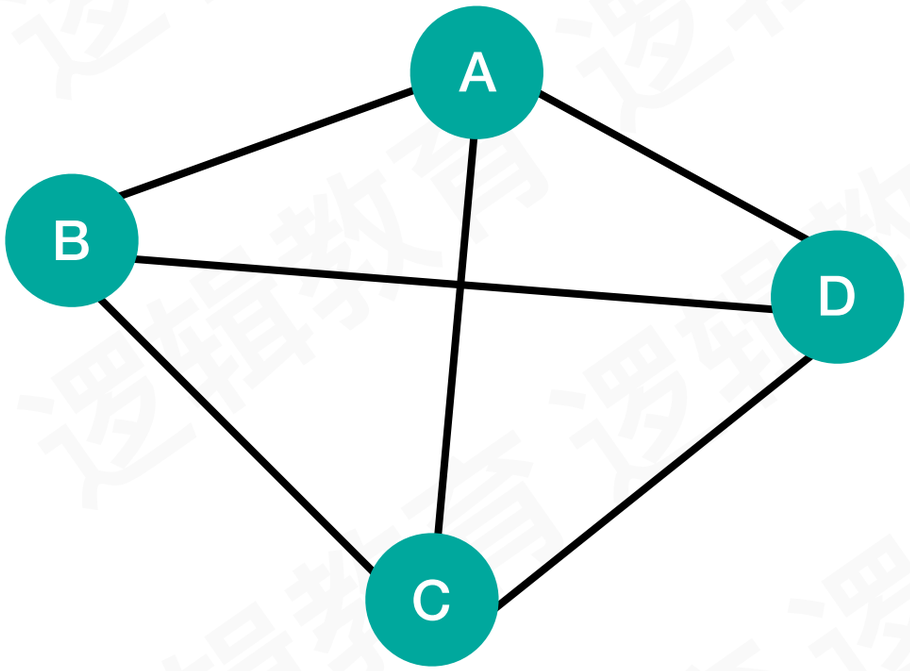
无向图 & 无向边



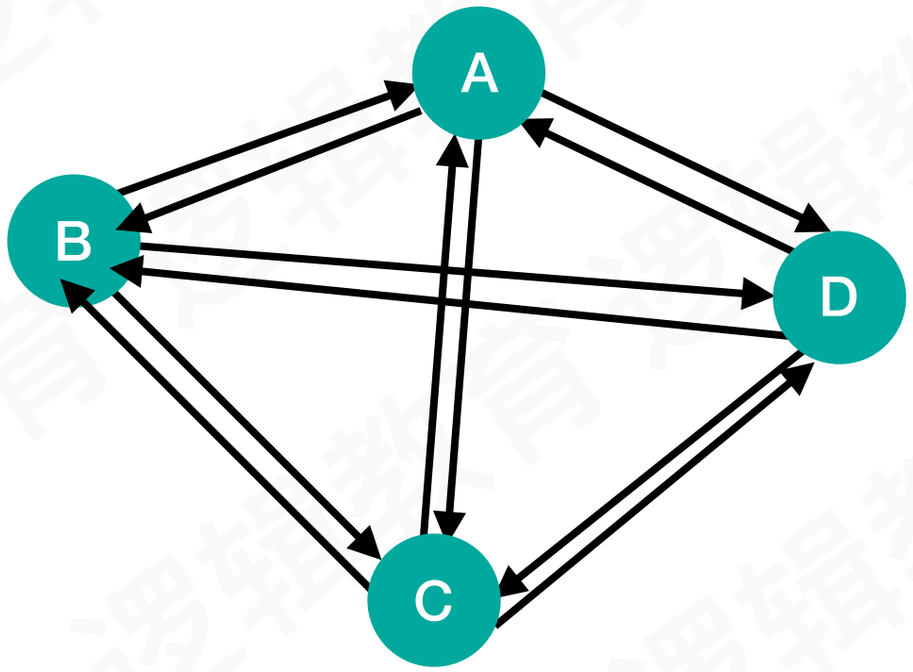
有向图 &有向边

课程研发:CC老师
课程授课:CC老师

各种图的定义(02)



无向完全图

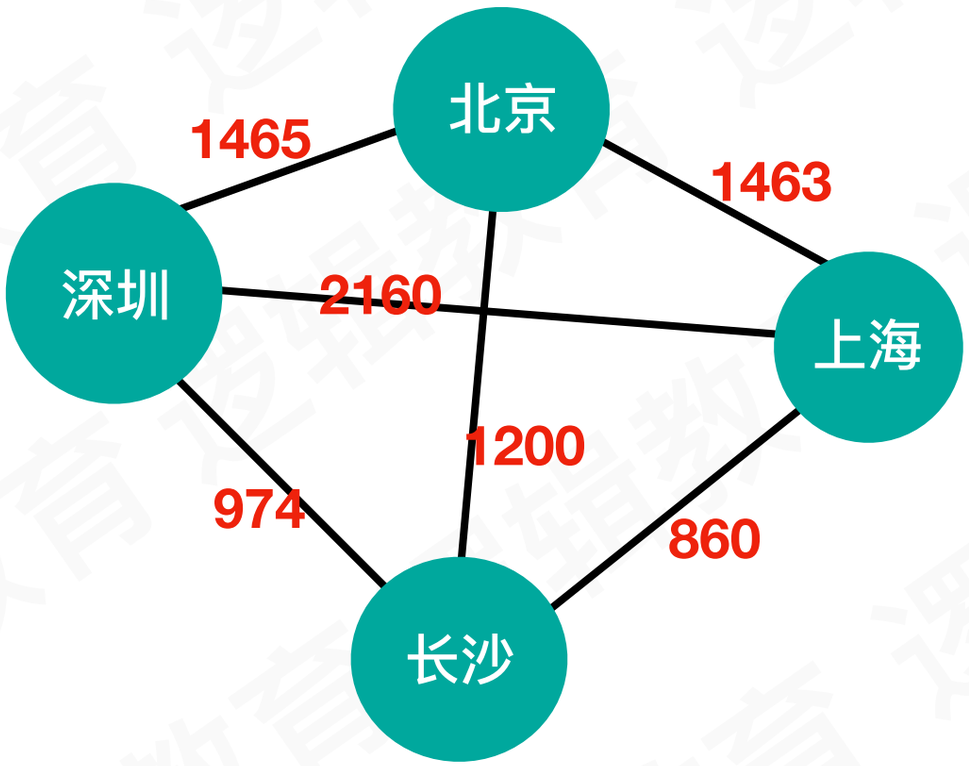


有向完全图

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

各种图的定义(03)

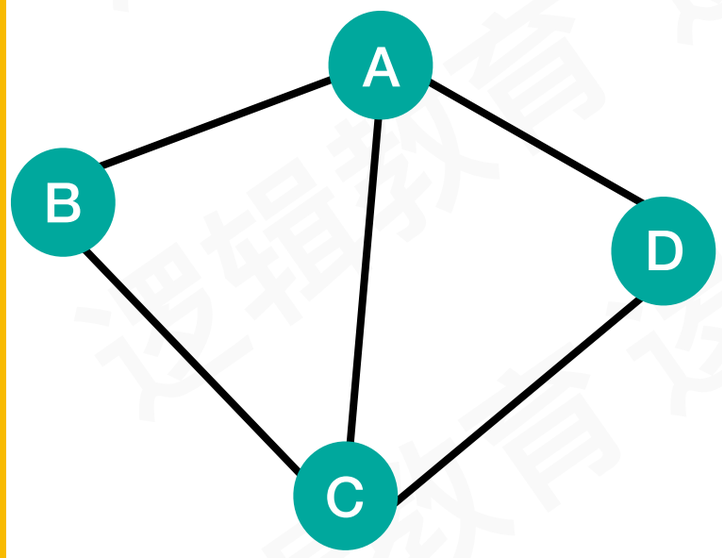


网

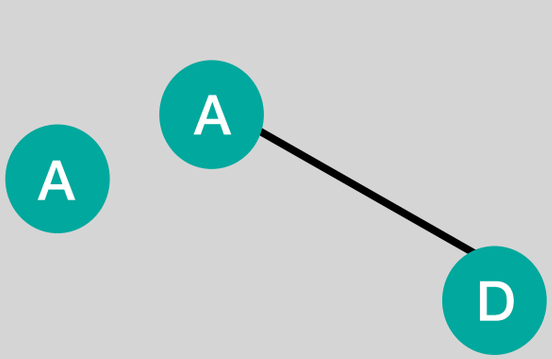
课程研发:CC老师
课程授课:CC老师

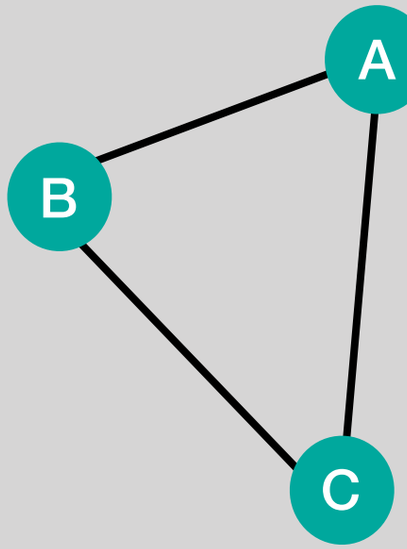
逻辑教育
Logic education

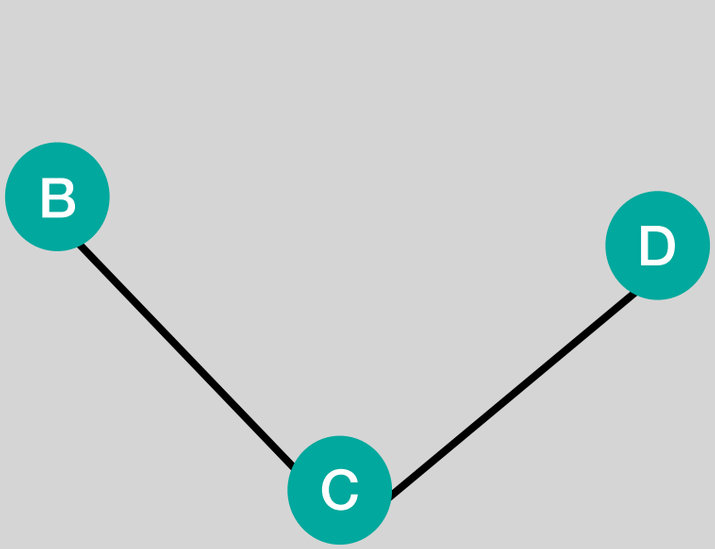
各种图的定义(03)



无向图G



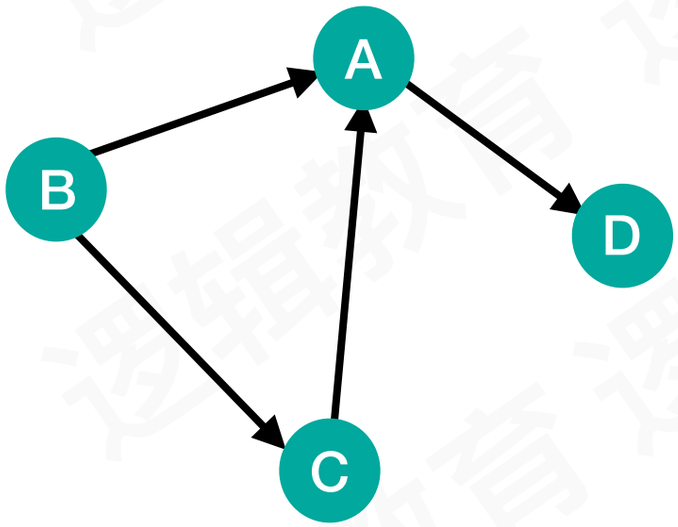




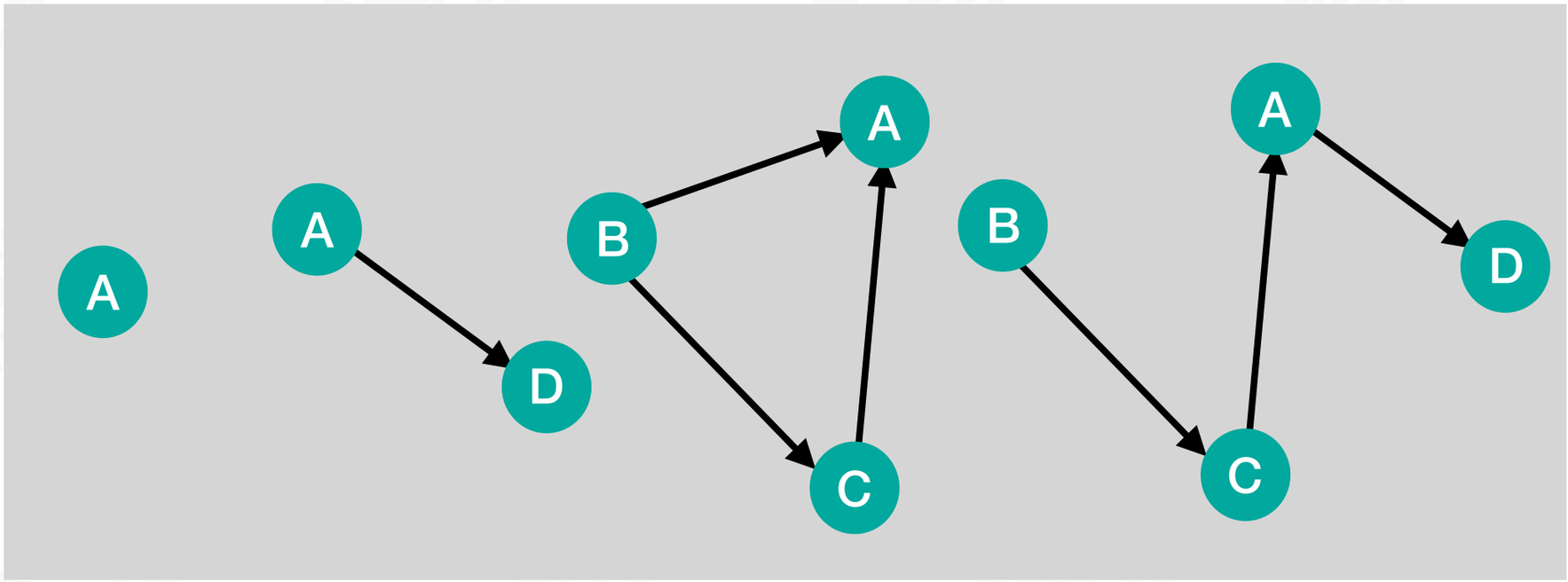
无向图G 的子图
(SubGraph)

课程研发:CC老师
课程授课:CC老师

各种图的定义(03)



有向图G



有向图G 的子图
(SubGraph)

逻辑教育
Logic education

各种图的定义(04)

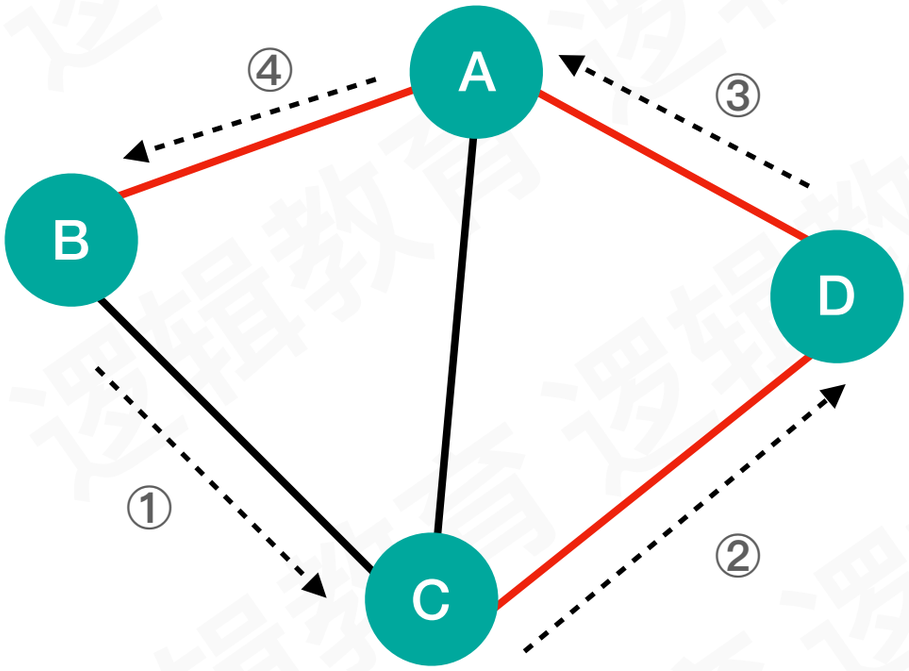


图1

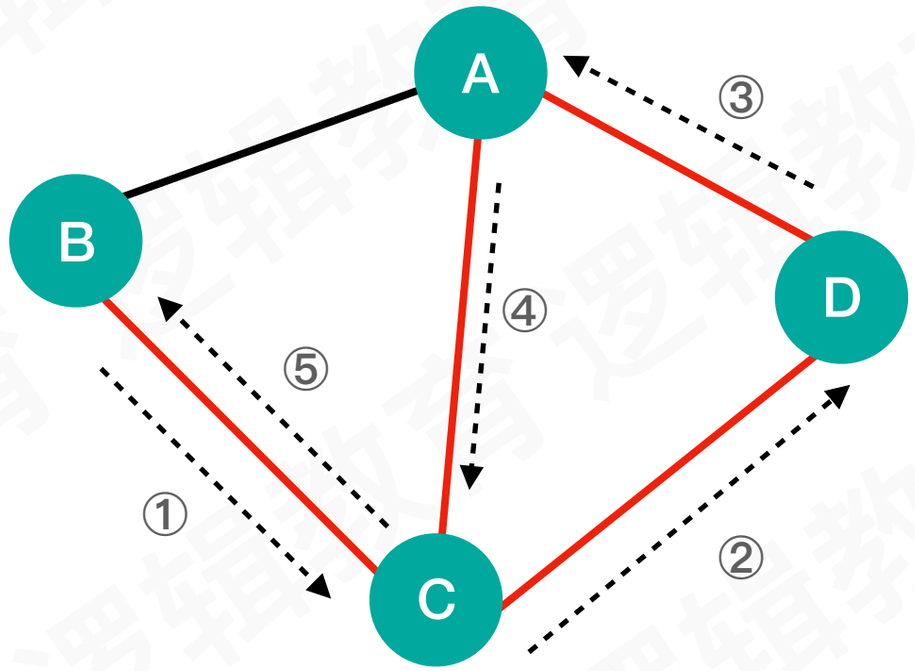


图2

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

各种图的定义(05)

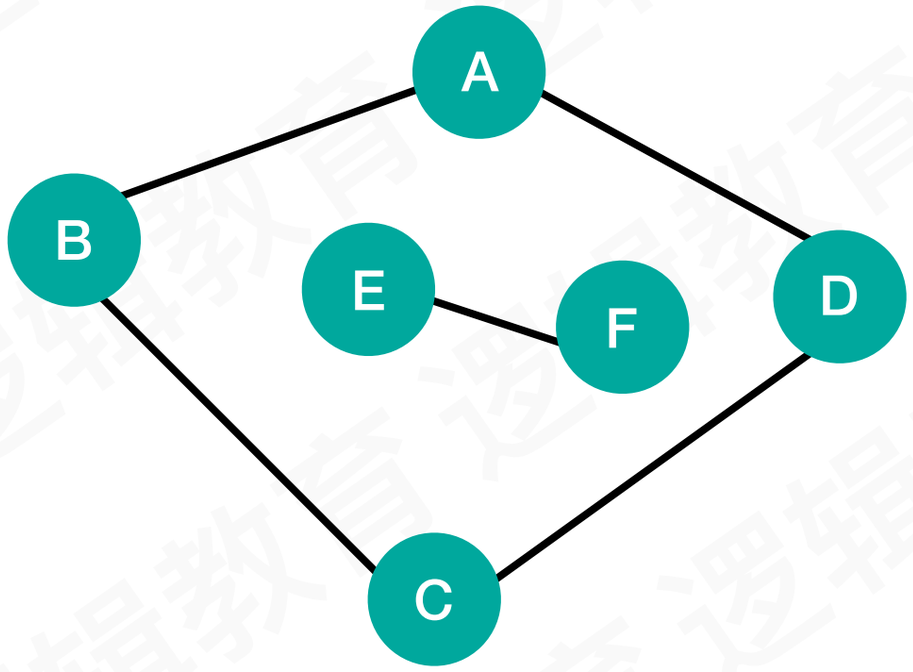


图1

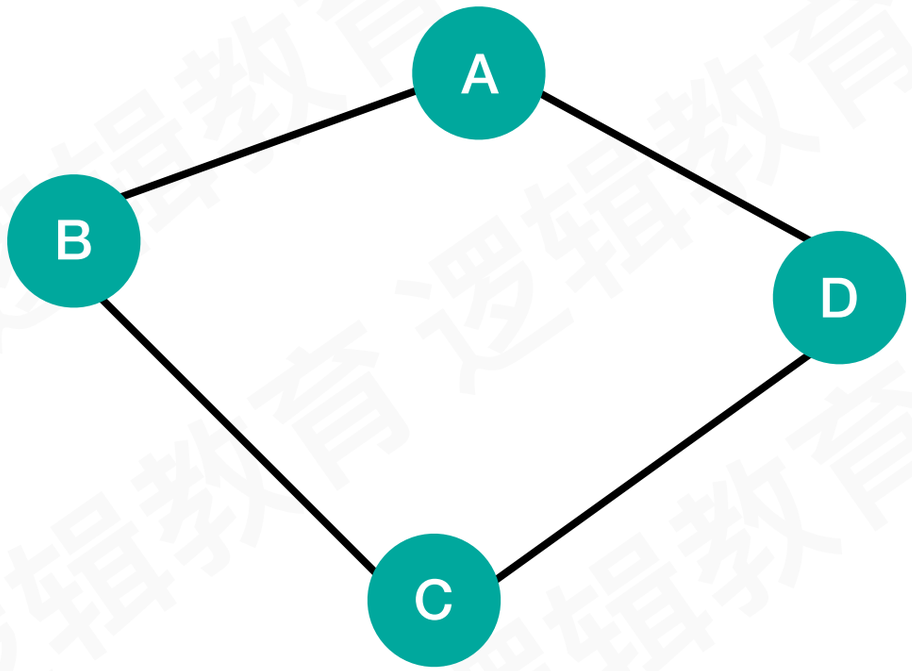


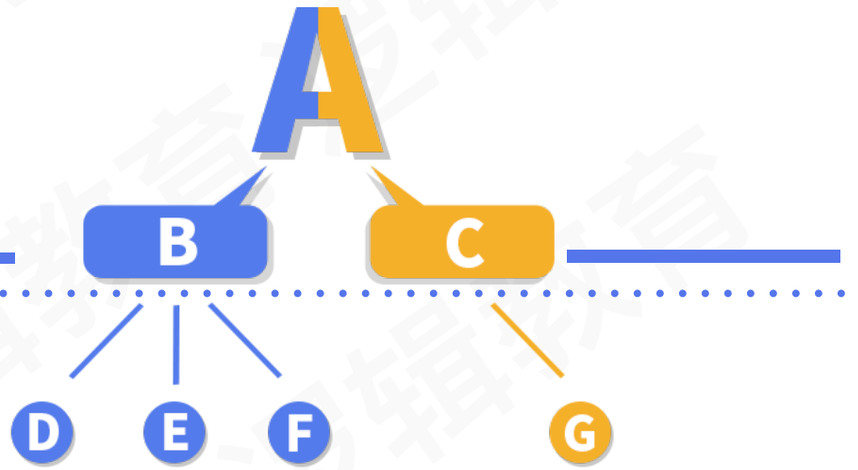
图2

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Hello 数据结构与算法

图的应用-图的存储



数据结构与算法主题

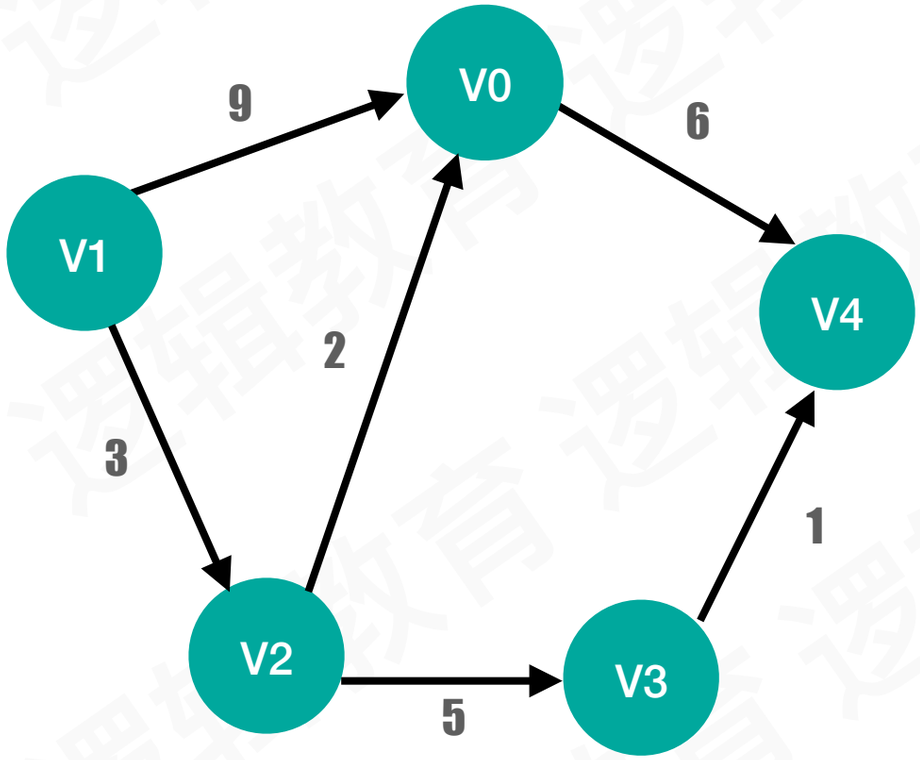
@CC老师

全力以赴．非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的存储思考



【数据结构与算法设计】 将左边图存储到计算机中.请设计一个数据结构并将其合理存储起来. **#快手面试真题#**

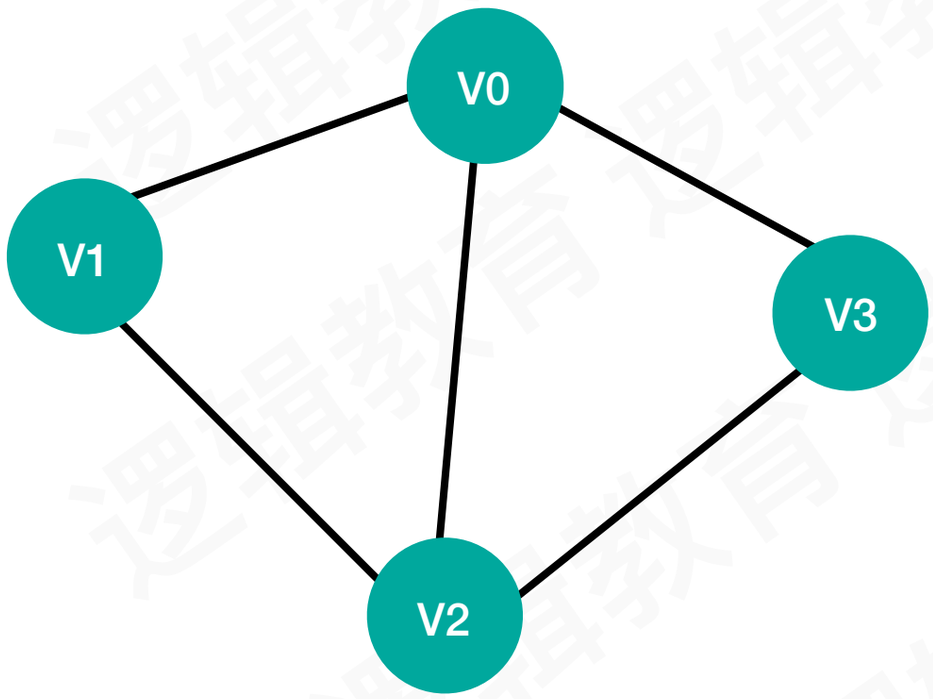
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

邻接矩阵

设图G 有n个顶点,则邻接矩阵是一个n * n 的方阵 定义为:

$$arc[i][j] = \begin{cases} 1, 若(v_i, v_j) \in E 或者 <v_i, v_j> \in E \\ 0, 反之 \end{cases}$$



边数组:

顶点数组: V0 V1 V2 V3

主对角线

	V_0	V_1	V_2	V_3
V_0	0	1	1	1
V_1	1	0	1	0
V_2	1	1	0	1
V_3	1	0	1	0

V_1 的度为2

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

邻接矩阵思考

顶点数组: V0 V1 V2 V3

主对角线

边数组:

	V_0	V_1	V_2	V_3
V_0	0	1	1	1
V_1	1	0	1	0
V_2	1	1	0	1
V_3	1	0	1	0

V_1 的度为2

求得顶点V2的邻接点,为1的则是

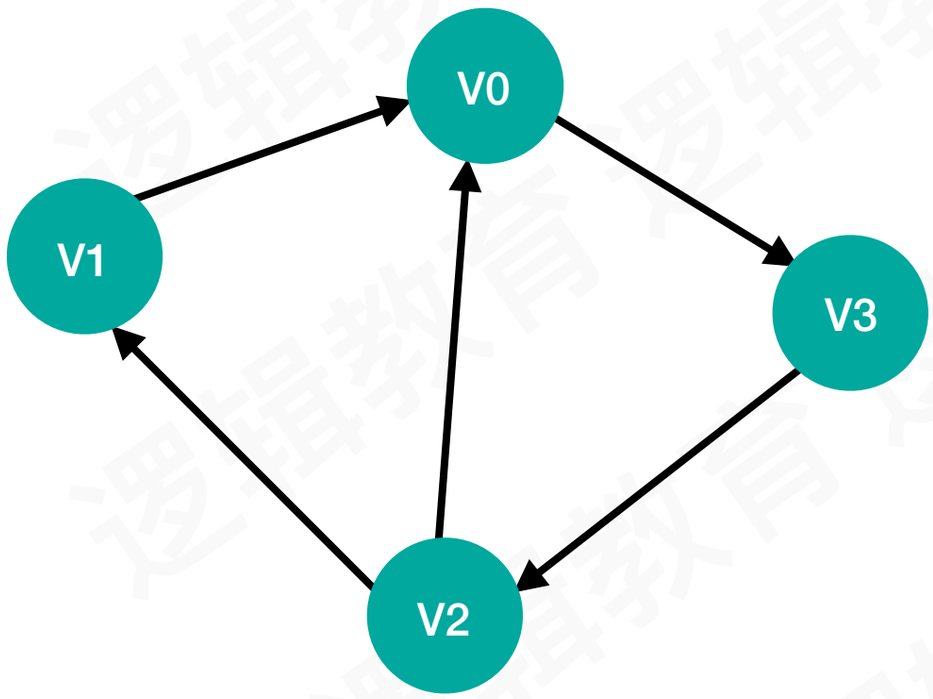
课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

邻接矩阵

设图G 有n个顶点,则邻接矩阵是一个n * n 的方阵 定义为:

$$arc[i][j] = \begin{cases} 1, 若(v_i, v_j) \in E 或者 <v_i, v_j> \in E \\ 0, 反之 \end{cases}$$



顶点数组: V0 V1 V2 V3

主对角线

边数组:

	V_0	V_1	V_2	V_3
V_0	0	0	0	1
V_1	1	0	0	0
V_2	1	1	0	0
V_3	0	0	1	0

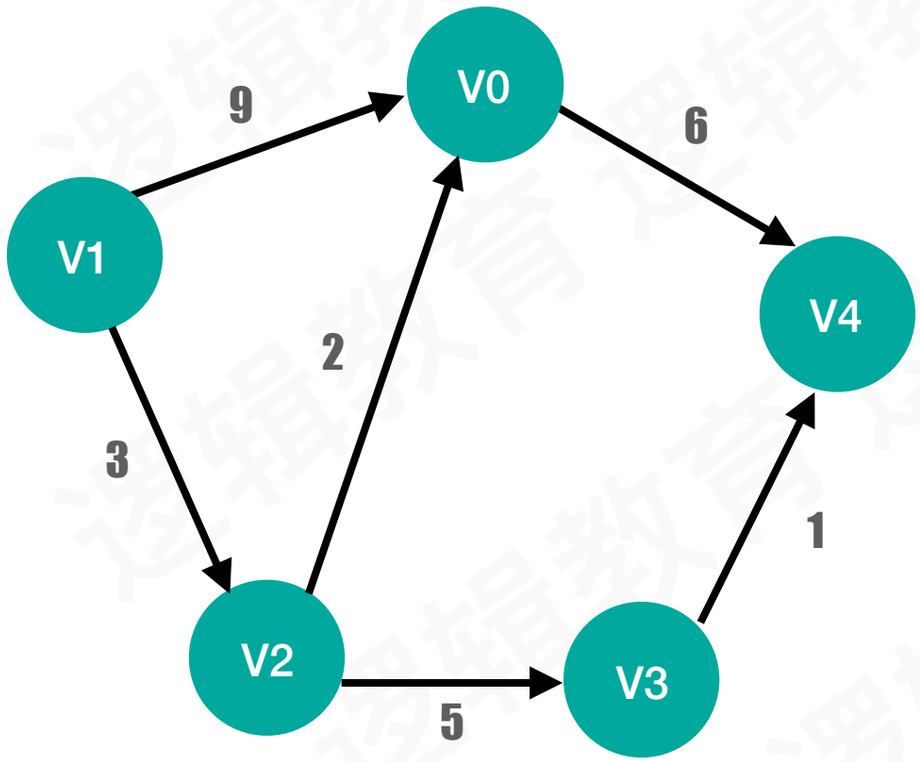
V_1 的度为1

逻辑教育
Logic education

邻接矩阵

设图G 是网图. 有n个顶点, 则邻接矩阵是一个n * n 的方阵 定义为:

$$arc[i][j] = \begin{cases} w_{ij}, 若(v_i, v_j) \in E 或者 <v_i, v_j> \in E \\ 0, 若 i=j \\ \infty \quad 反之 \end{cases}$$



顶点数组: V0 V1 V2 V3 V4

	V_0	V_1	V_2	V_3	V_4
V_0	0	∞	∞	∞	6
V_1	9	0	3	∞	∞
V_2	2	∞	0	5	∞
V_3	∞	∞	∞	0	1
V_4	∞	∞	∞	∞	0

课程研发:CC老师
课程授课:CC老师

邻接矩阵矩阵存储的数据结构设计

```
#define MAXVEX 100 /* 最大顶点数，应由用户定义 */
#define INFINITY 65535 /* 用65535表示∞ */

typedef int Status;  /* Status是函数的类型,其值是函数结果状态代码，如OK等 */
typedef char VertexType; /* 顶点类型应由用户定义  */
typedef int EdgeType; /* 边上的权值类型应由用户定义 */
typedef struct
{
    VertexType vexs[MAXVEX]; /* 顶点表 */
    EdgeType arc[MAXVEX][MAXVEX];/* 邻接矩阵，可看作边表 */
    int numNodes, numEdges; /* 图中当前的顶点数和边数  */
}MGraph;
```

课程研发:CC老师
课程授课:CC老师

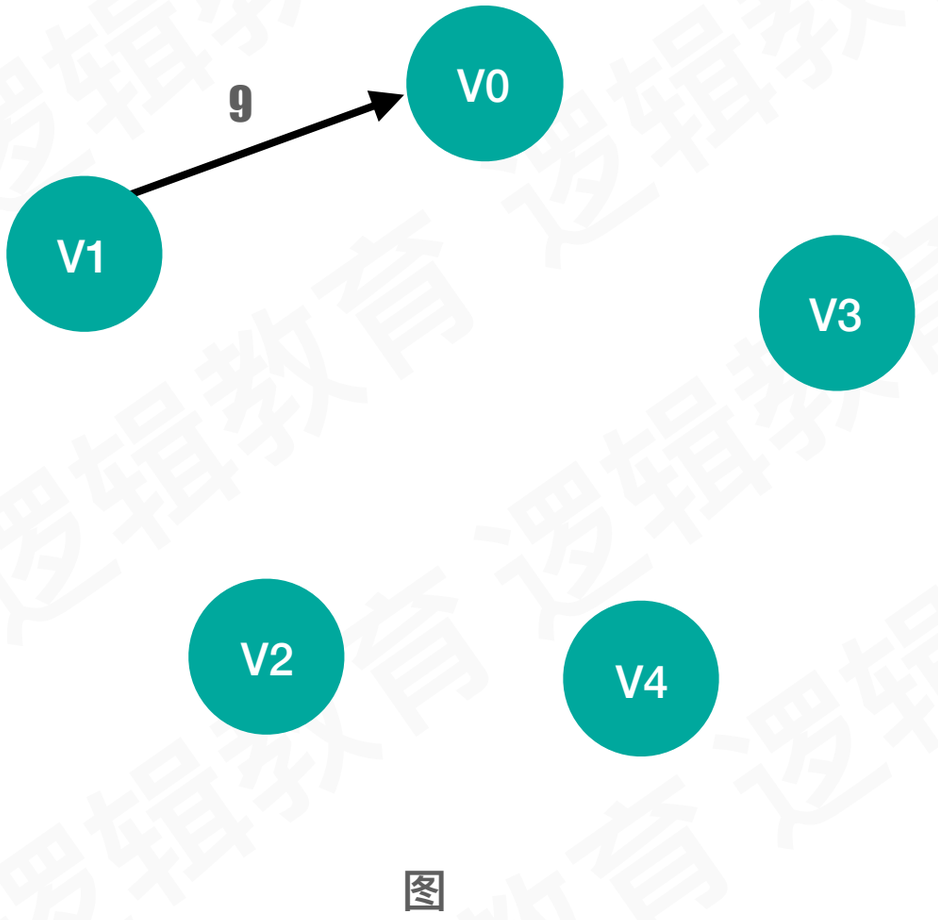
逻辑教育
Logic education

邻接矩阵矩阵存储代码实现思路

1. 确定顶点数/边数
2. 读取顶点信息
3. 初始化邻接矩阵
4. 读入边信息
5. 循环打印

课程研发:CC老师
课程授课:CC老师

邻接矩阵



图

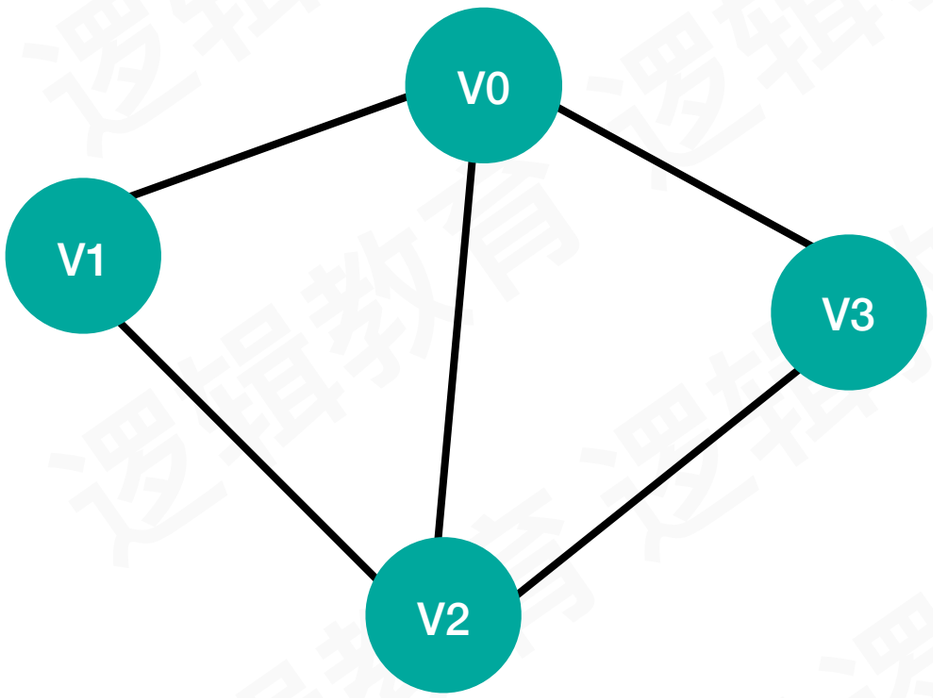
顶点数组: V0 V1 V2 V3 V4

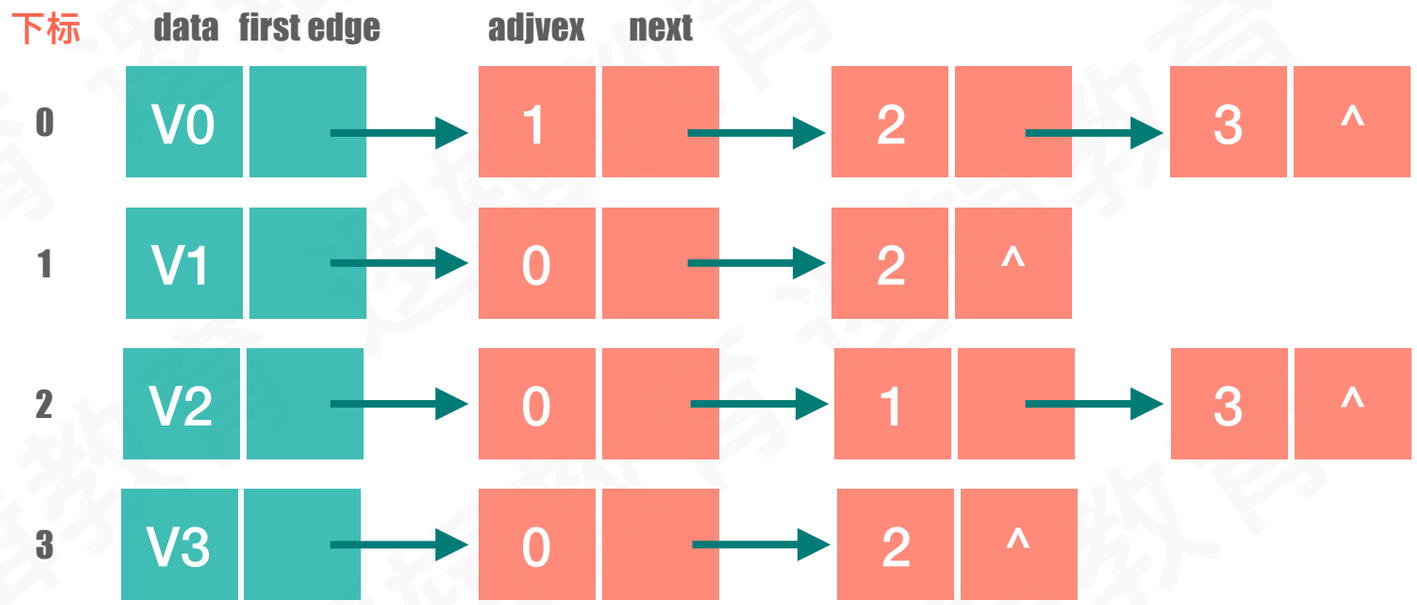
	V_0	V_1	V_2	V_3	V_4
V_0	0	∞	∞	∞	∞
V_1	9	0	∞	∞	∞
V_2	∞	∞	0	∞	∞
V_3	∞	∞	∞	0	∞
V_4	∞	∞	∞	∞	0

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

邻接表

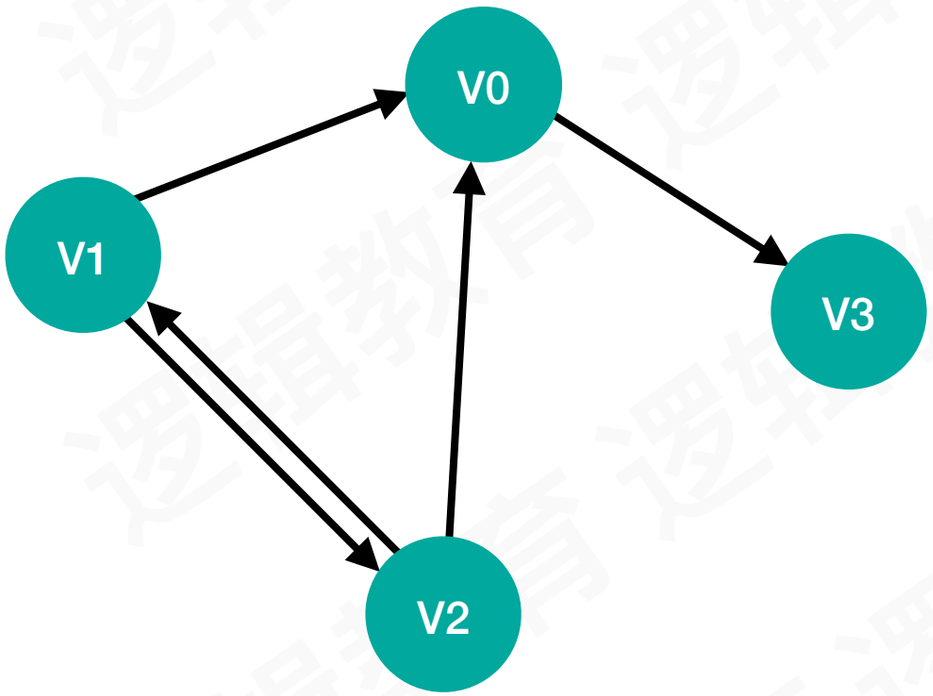


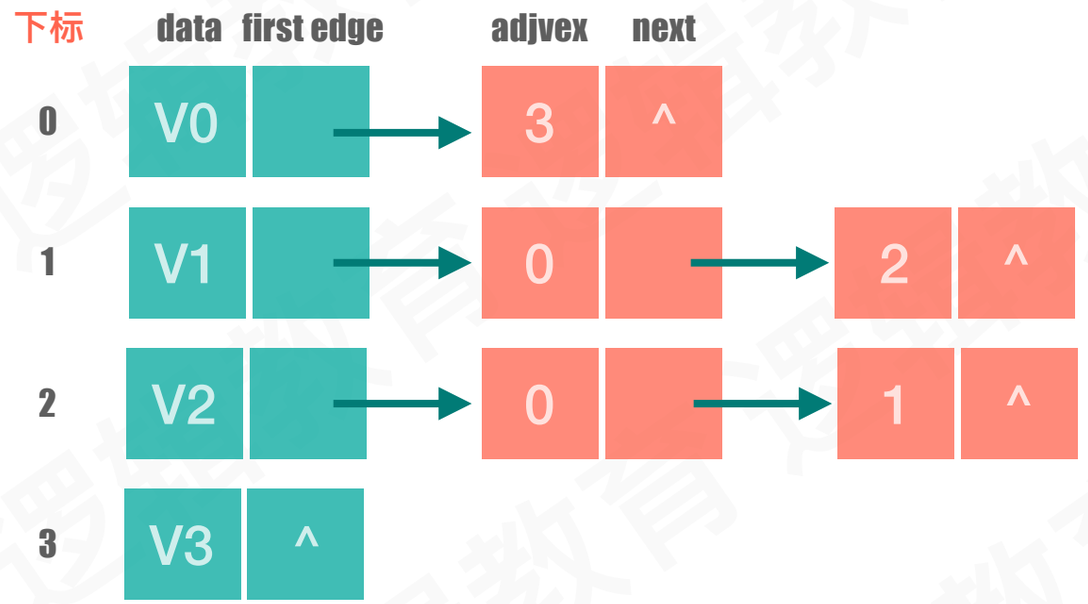


课程研发:CC老师
课程授课:CC老师

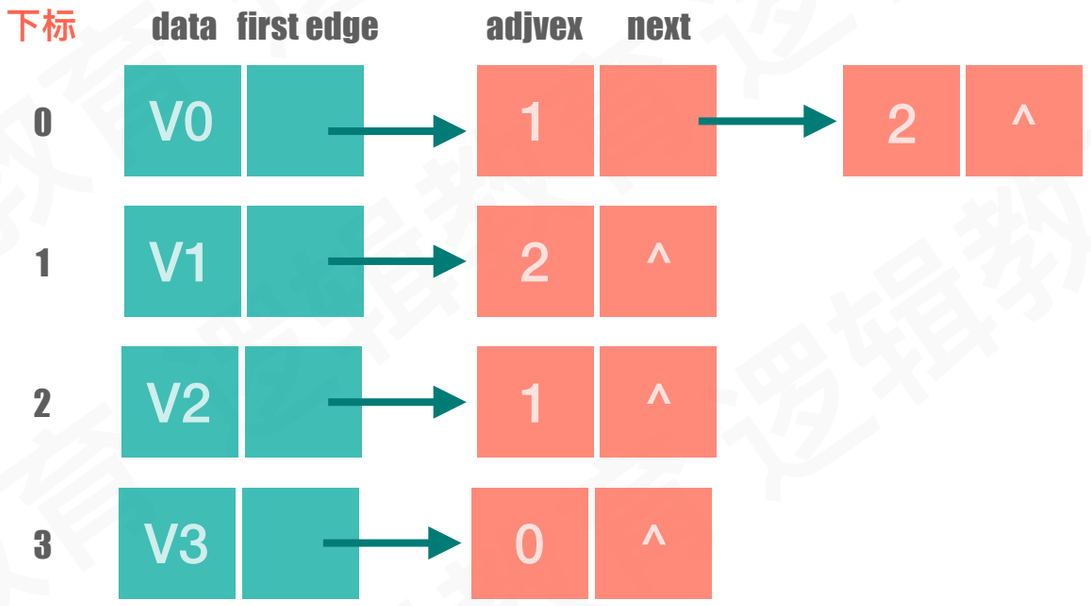
逻辑教育
Logic education

邻接表





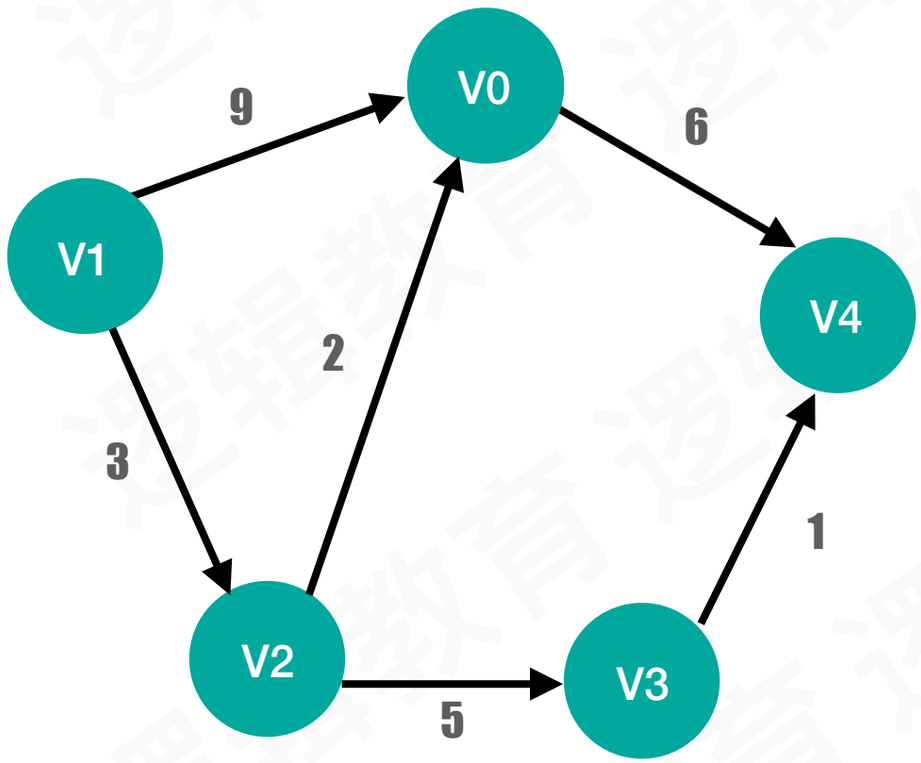
有向图邻接表

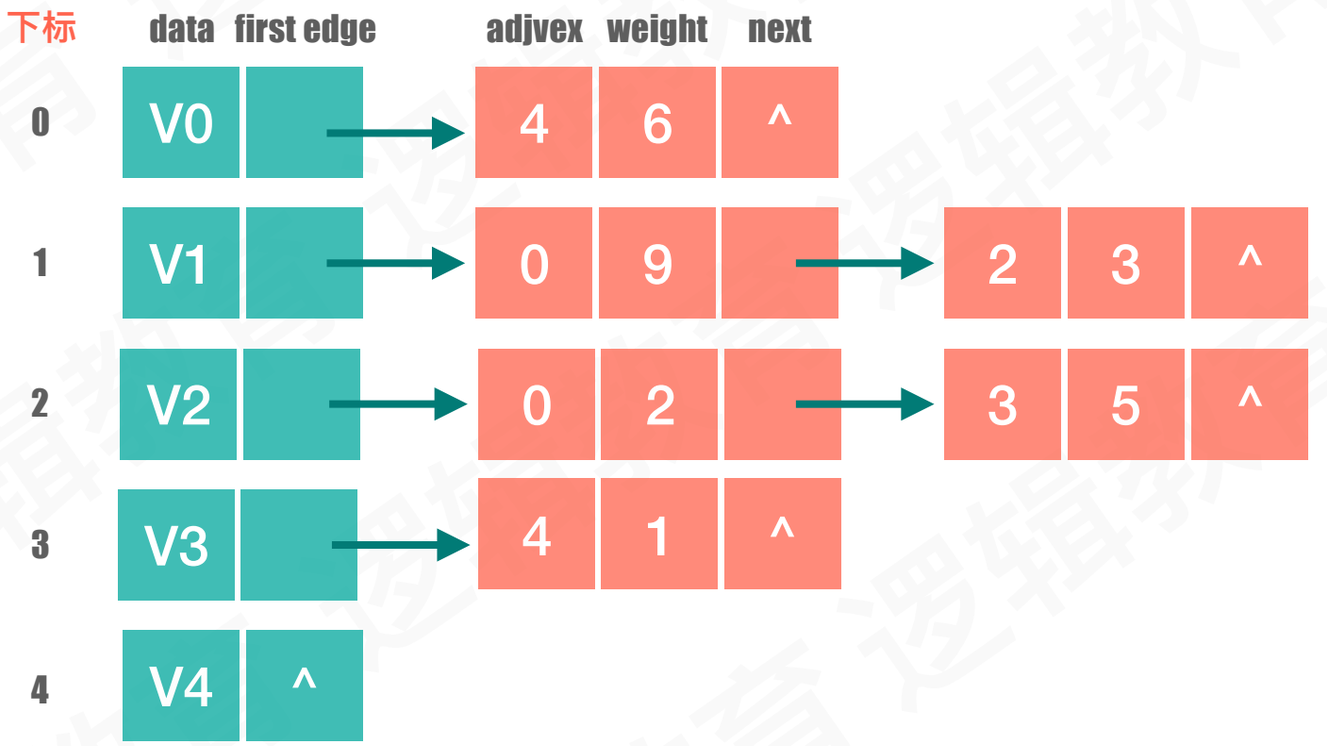


有向图逆邻接表

课程研发:CC老师
课程授课:CC老师

邻接表





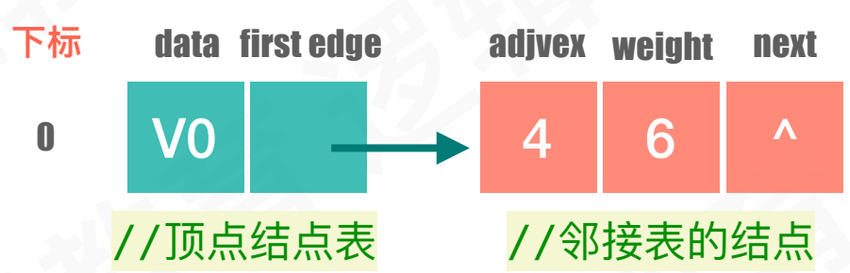
课程研发:CC老师
课程授课:CC老师

邻接表存储的数据结构设计

```
//邻接表的结点
typedef struct Node{
    int adj_vex_index;  //弧头的下标，也就是被指向的下标
    Element data;       //权重值
    struct Node * next; //边指针
}EdgeNode;

//顶点结点表
typedef struct vNode{
    Element data;           //顶点的权值
    EdgeNode * firstedge;   //顶点下一个是谁?
}VertexNode, Adjlist[M];

//总图的一些信息
typedef struct Graph{
    Adjlist adjlist;        //顶点表
    int arc_num;            //边的个数
    int node_num;           //节点个数
    BOOL is_directed;       //是不是有向图
}Graph, *GraphLink;
```



课程研发:CC老师
课程授课:CC老师

邻接表存储的存储代码实现思路

1. 确定顶点数/边数

2. 读取顶点信息

3. 创建一个结点 插入到对应的顶点数组中

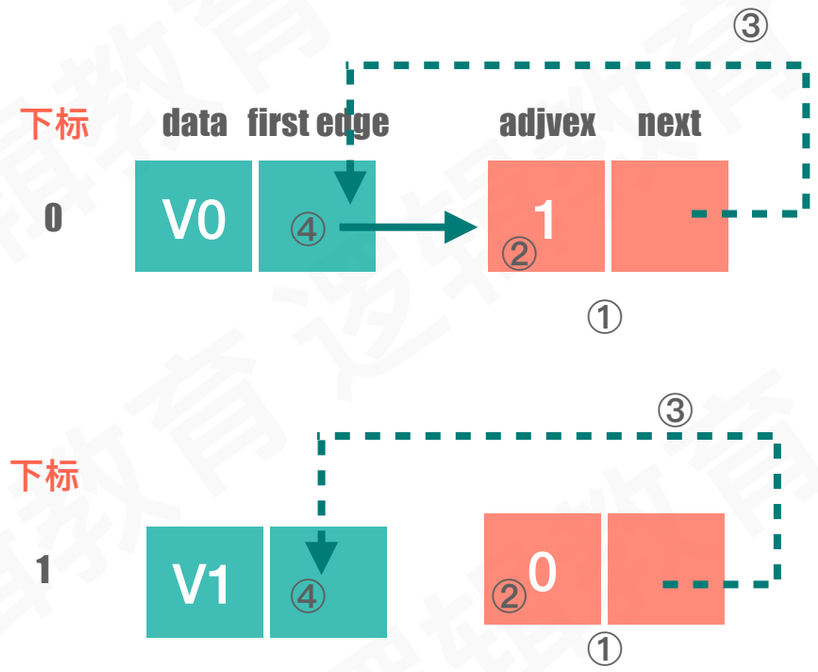
① 创建结点p

② 将结点p的adjvex 赋值 j

③ 将结点p 插入到对应的顶点数组下标i下

④ 将顶点数组[i] 的firstedge设置为p

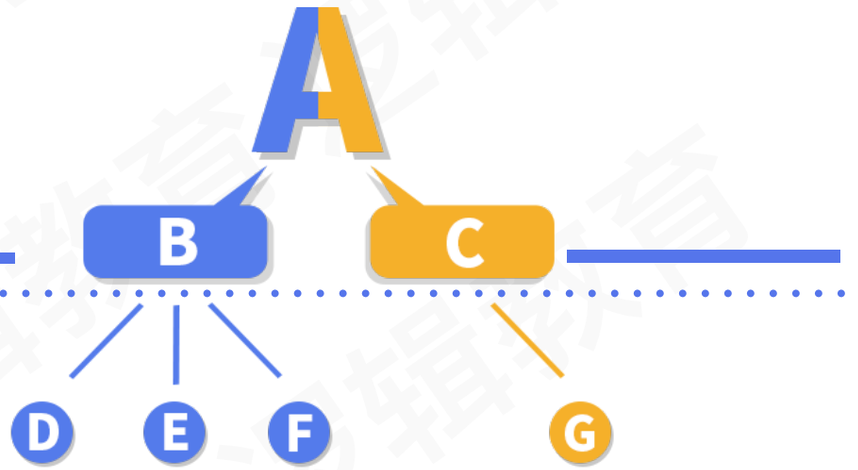
如果是无向图,则循环①~⑤步骤.



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Hello 数据结构与算法



图的应用-图的遍历

数据结构与算法主题

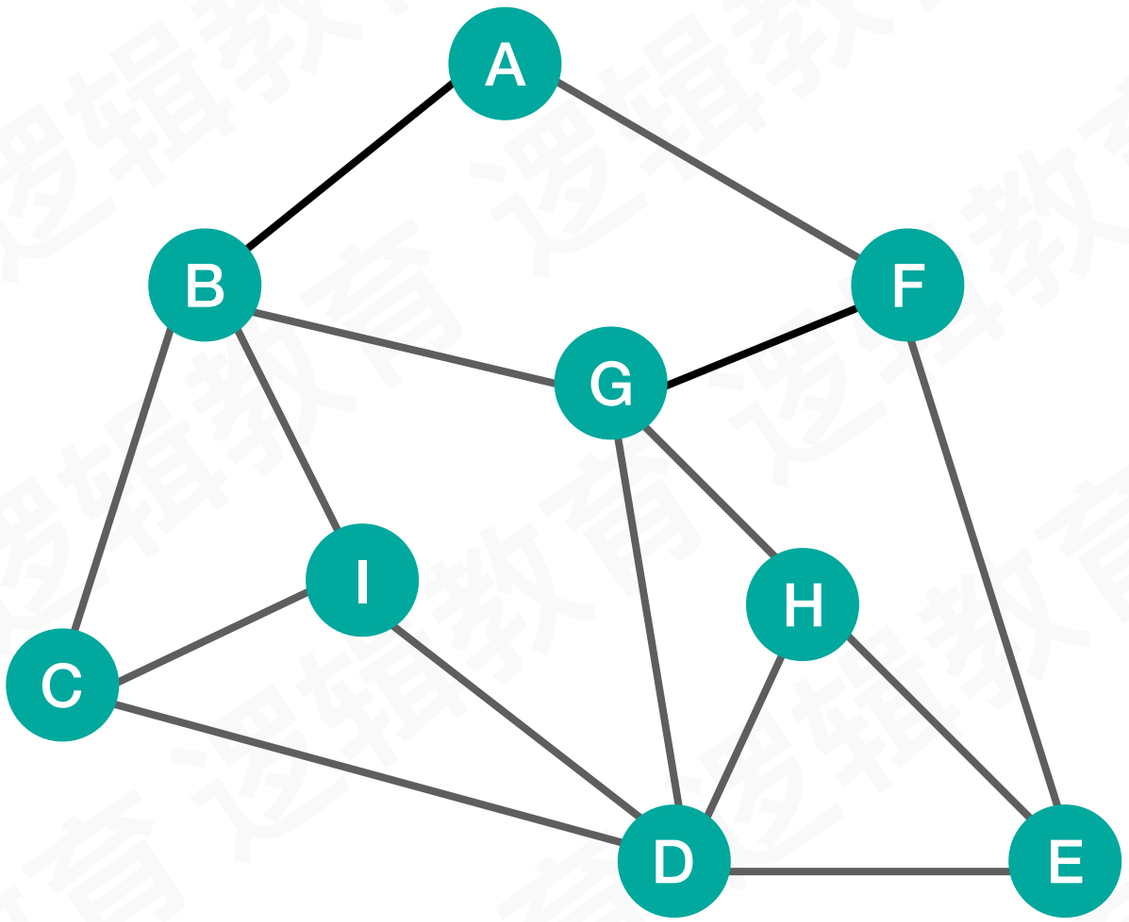
@CC老师

全力以赴，非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

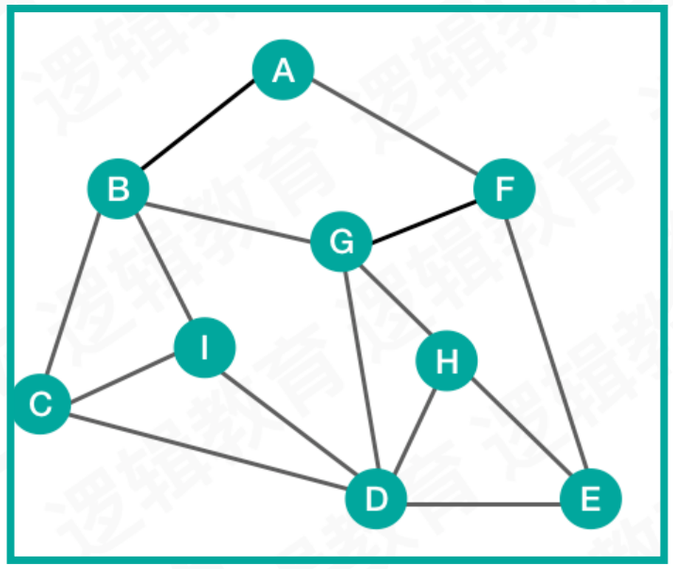
图的遍历

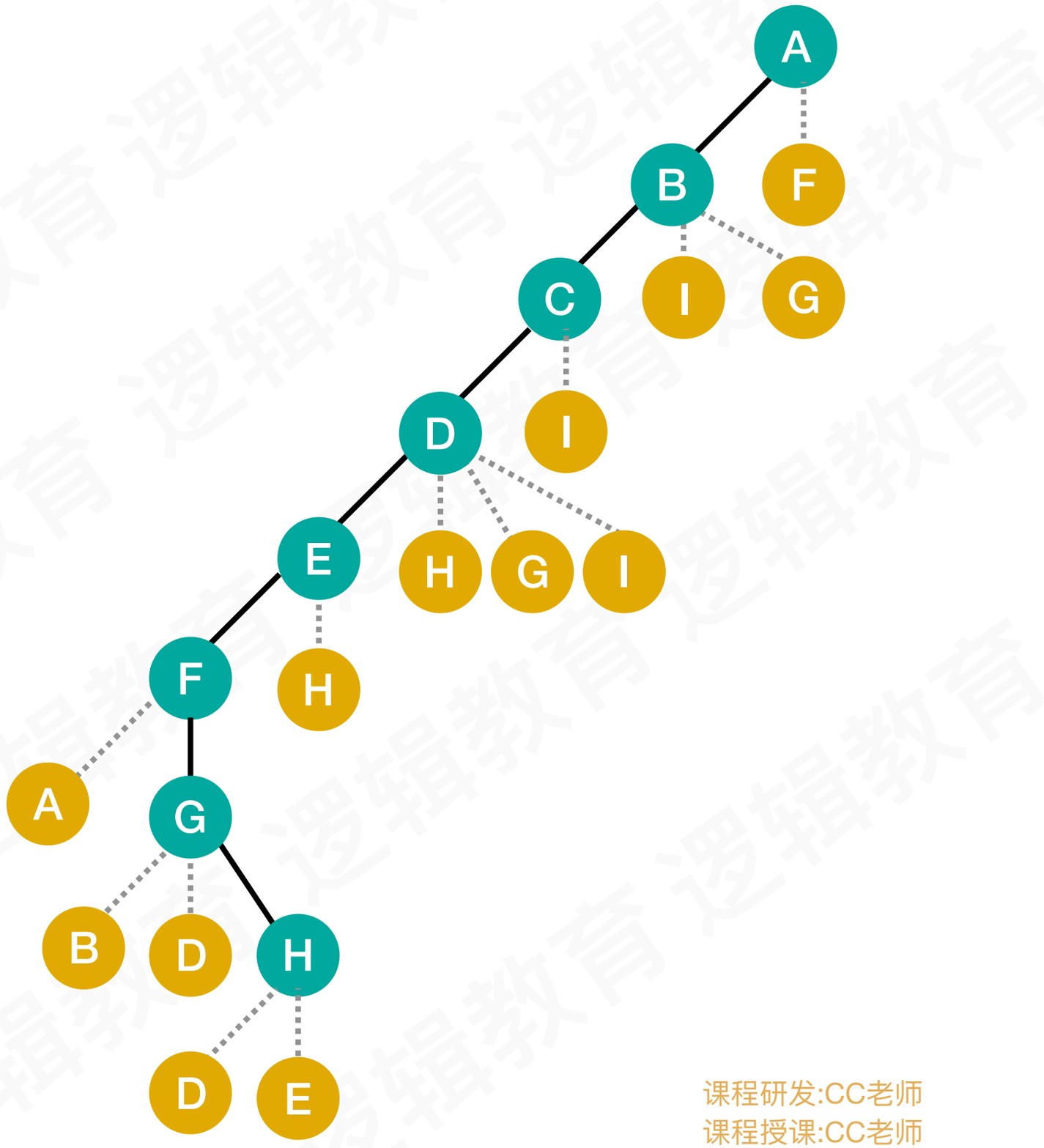


课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历

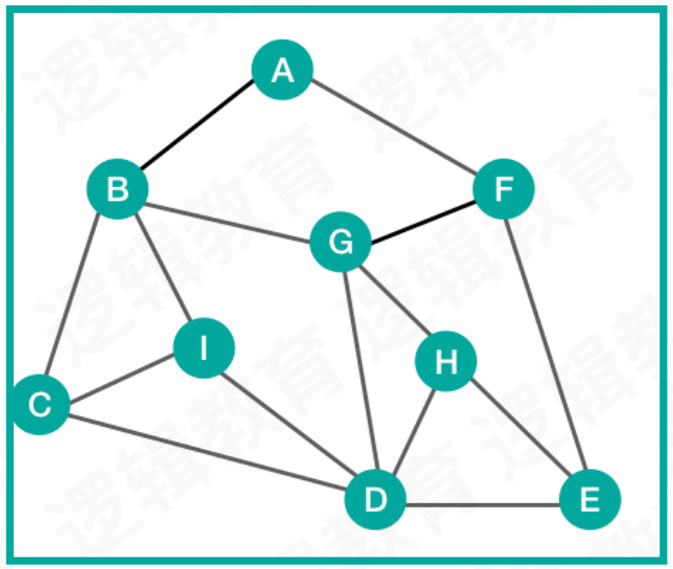


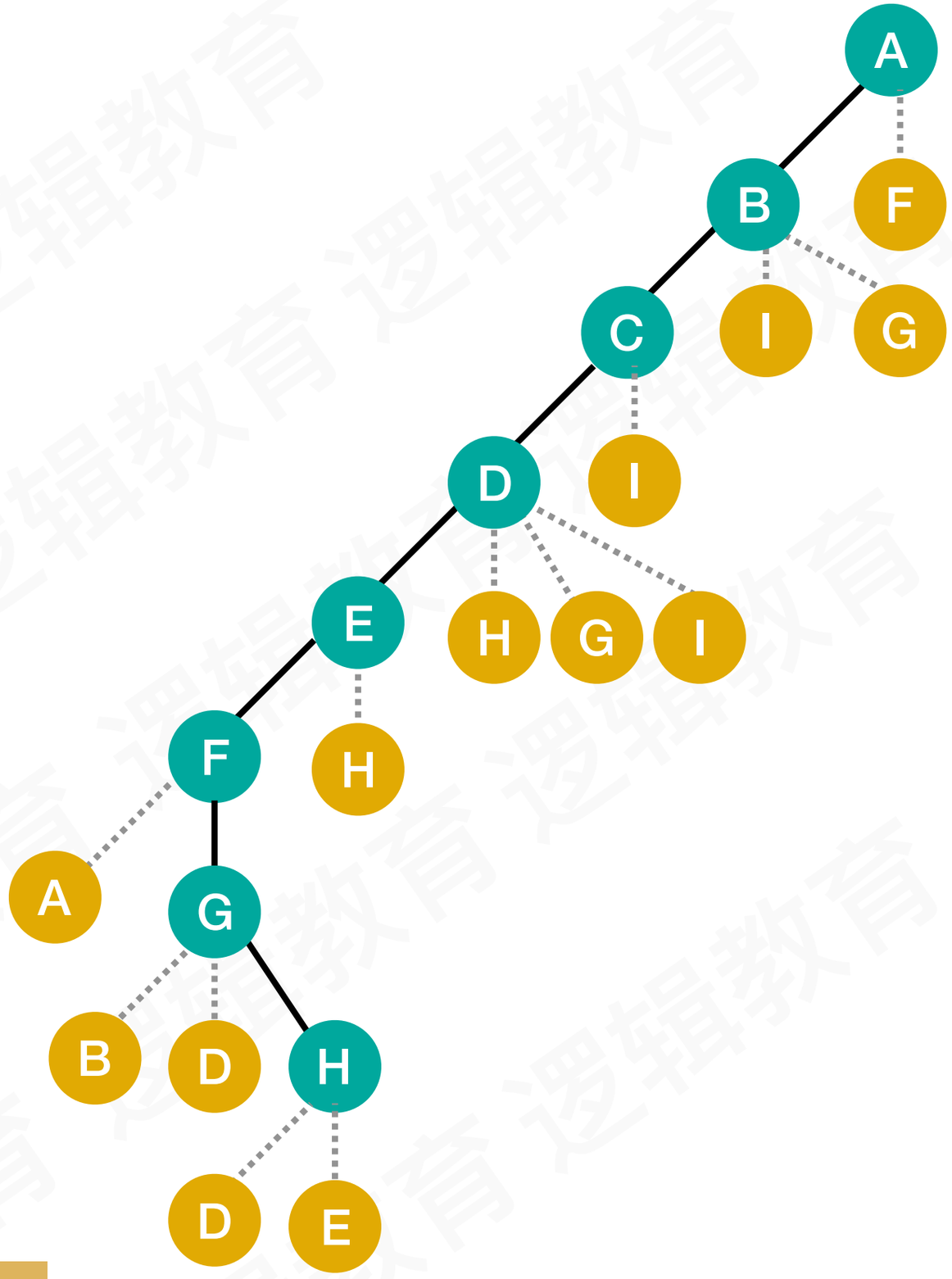


课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历



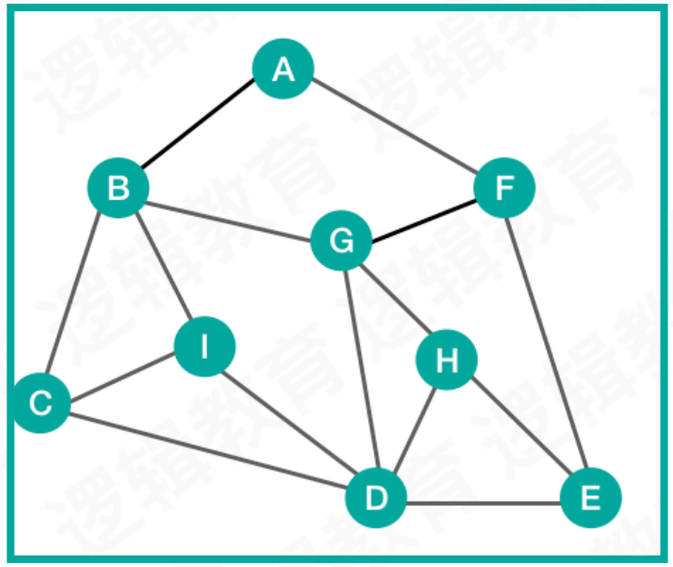


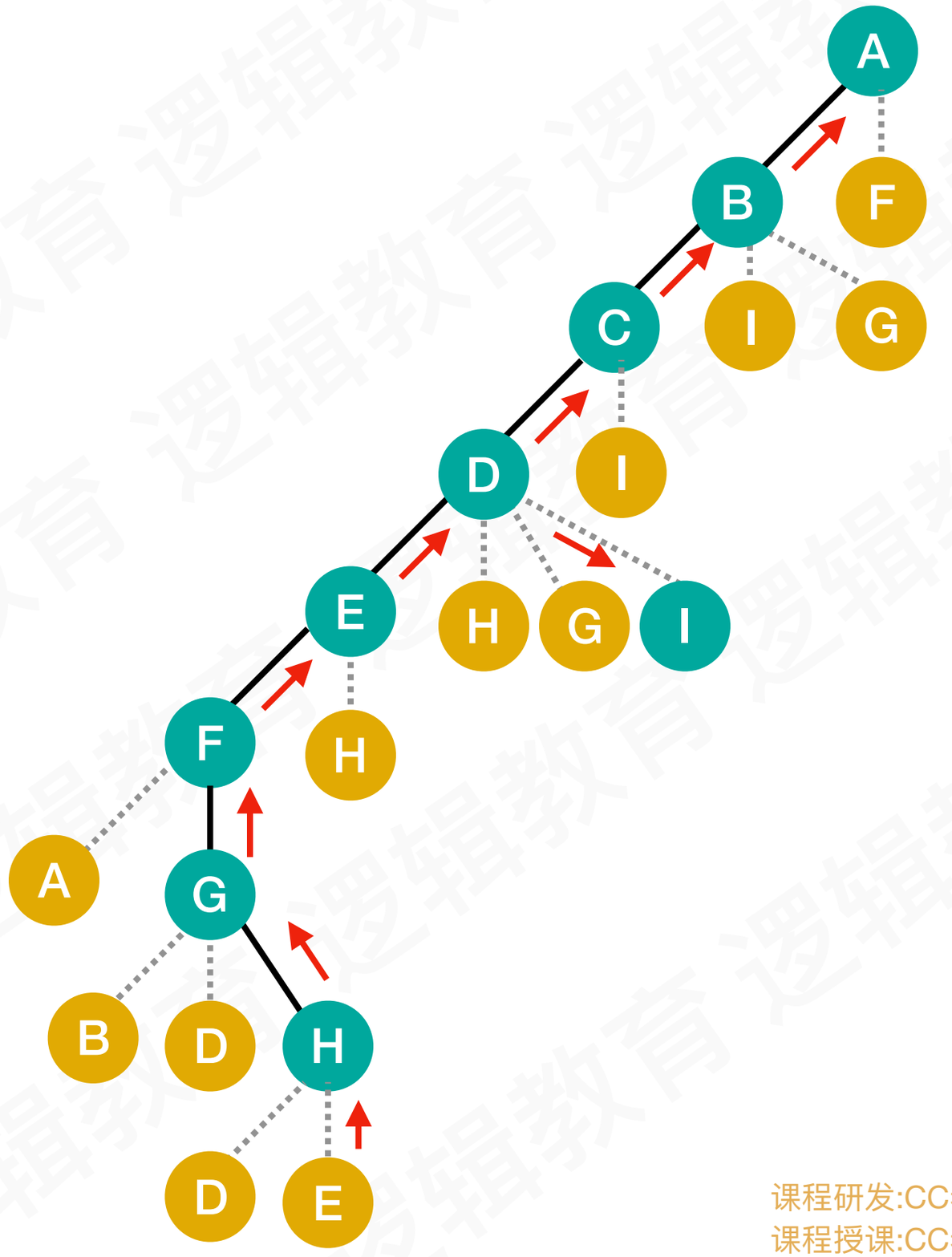
是否找到了所有的顶点了?

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历—深度优先遍历





课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

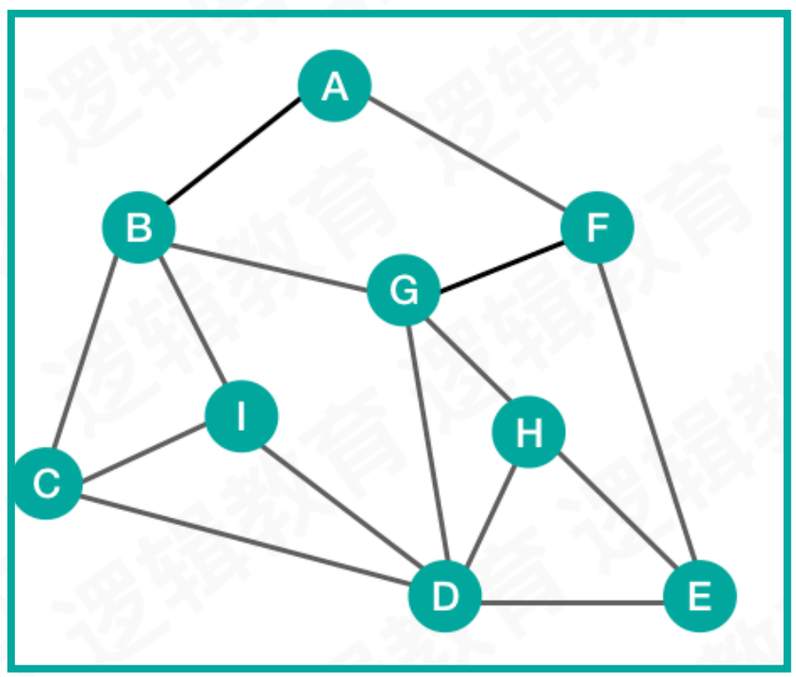
图的遍历—邻接矩阵深度优先遍历

顶点数组: A B C D E F G H I

主对角线 A0 B1 C2 D3 E4 F5 G6 H7 I8

边数组:

	A0	B1	C2	D3	E4	F5	G6	H7	I8
A0	0	1				1			
B1	1	0	1				1		1
C2		1	0	1					1
D3			1	0	1		1	1	1
E4				1	0	1		1	
F5	1				1	0	1		
G6		1		1		1	0	1	
H7				1	1		1	0	
I8		1	1	1					0



课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历—邻接矩阵深度优先遍历代码实现思路

1. 将图的顶点和边信息输入到图结构中;

2. 创建一个visited 数组,用来标识顶点是否已经被遍历过.

3. 初始化visited 数组,将数组中元素置为FALSE

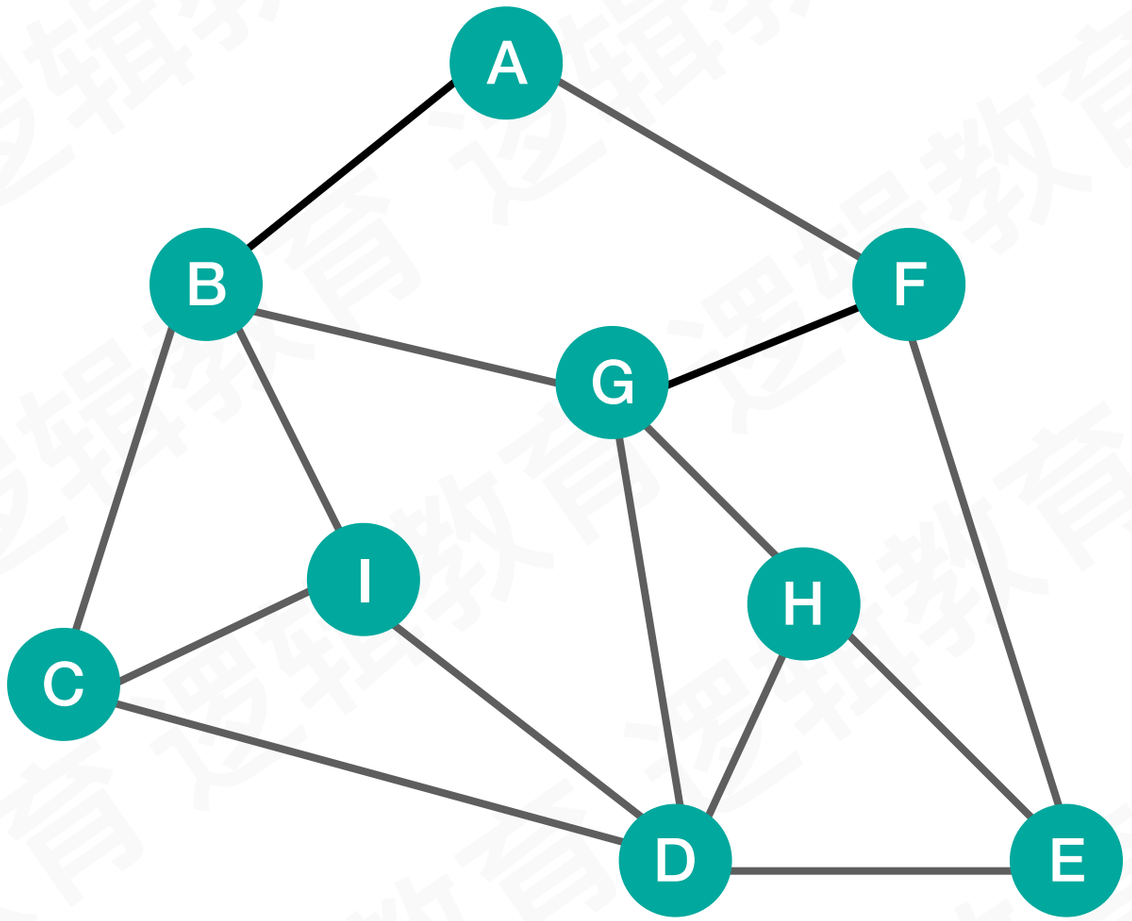
4. 选择顶点开始遍历.(注意非连通图的情况)

5. 进入递归; 打印i 对应的顶点信息. 并将该顶点标识为已遍历.

6. 循环遍历边表,判断当前arc[i][j] 是否等于1,并且当前该顶点没有被遍历过,则继续递归DFS;

课程研发:CC老师
课程授课:CC老师

图的遍历—邻接表对图的遍历处理

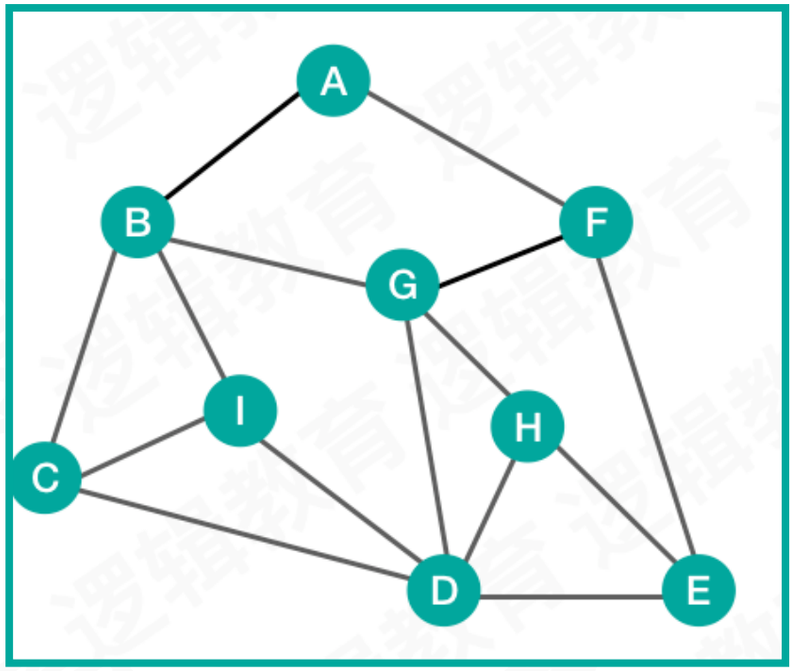


如何存储到邻接表中?

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

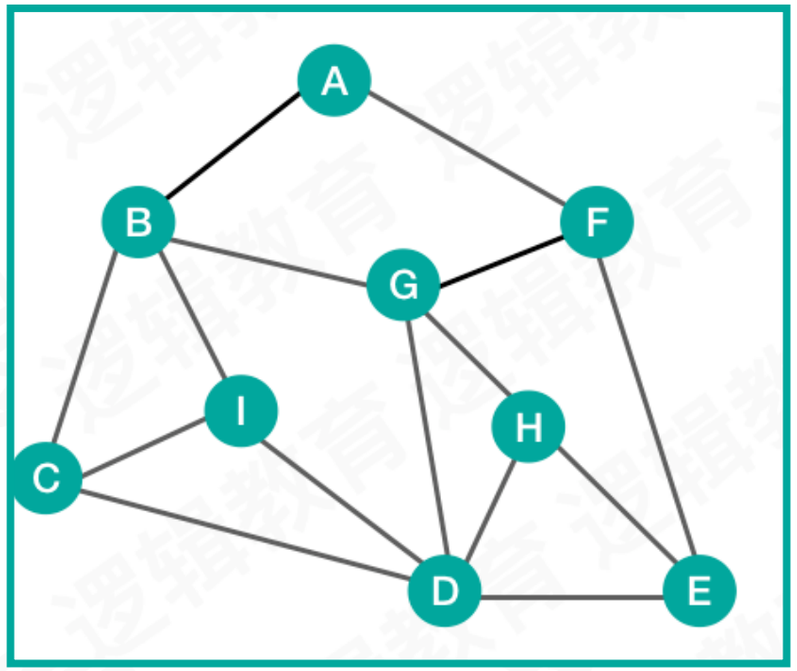
图的遍历—邻接表深度优先遍历

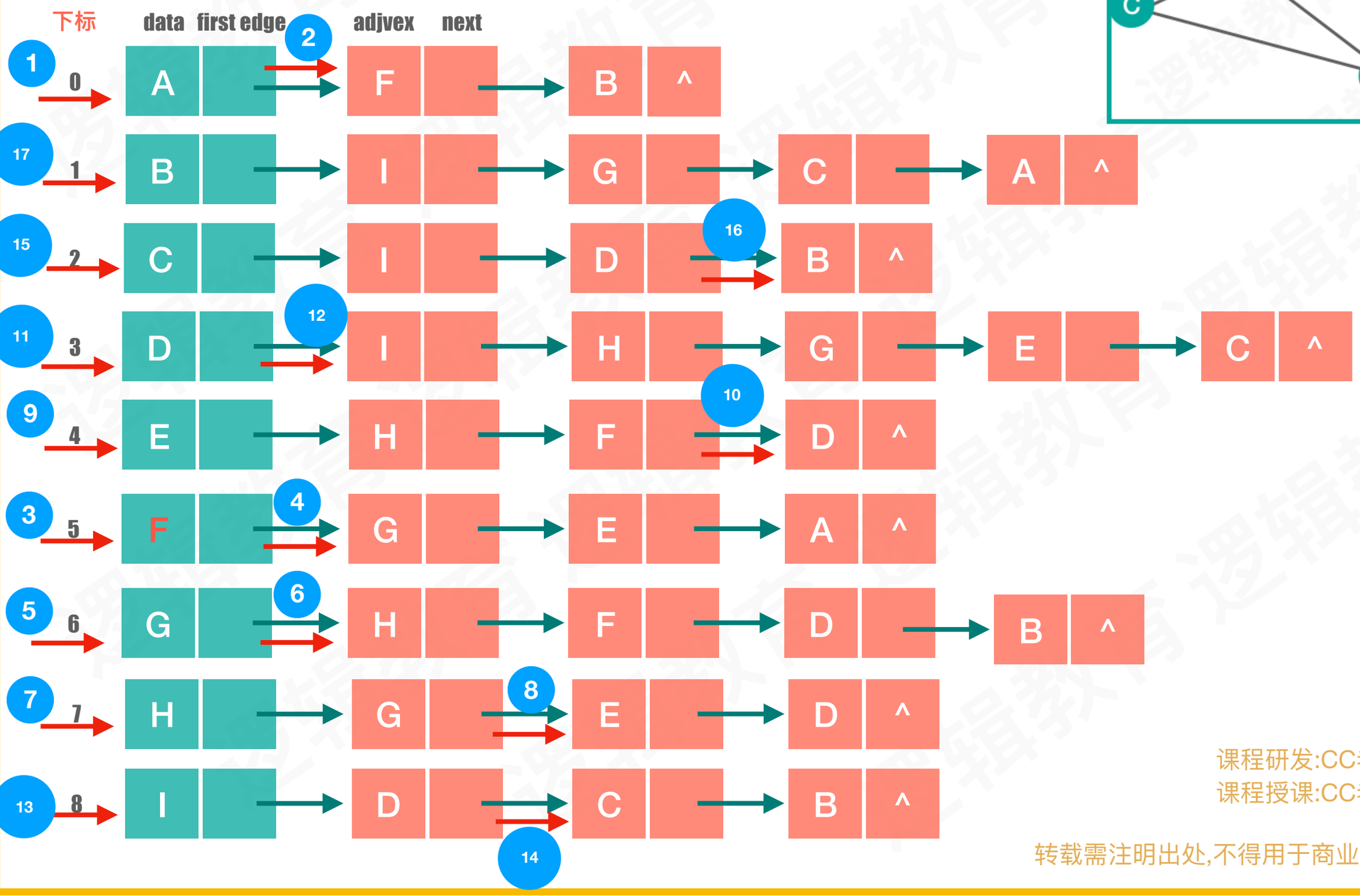


下标	data	first edge	adjvex	next	adjvex	next	adjvex	next
0	A	→	F		→	B	^	
1	B	→	I		→	G	→	C		→	A	^
2	C	→	I		→	D	→	B	^	
3	D	→	I		→	H	→	G	→	E	→	C	^
4	E	→	H		→	F	→	D	^	
5	F	→	G		→	E	→	A	^	
6	G	→	H		→	F	→	D	→	B	^
7	H	→	G		→	E	→	D	^	
8	I	→	D		→	C	→	B	^	

课程研发:CC老师
课程授课:CC老师

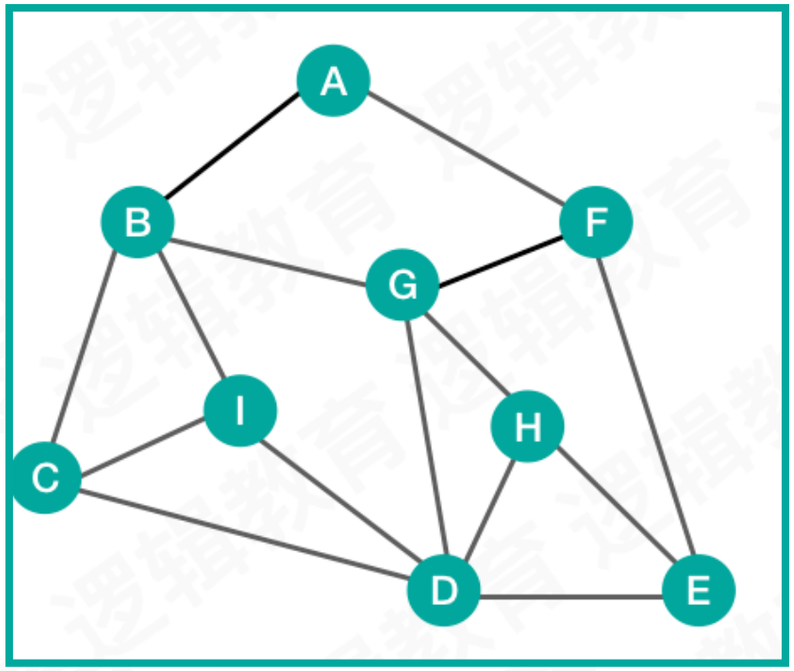
图的遍历—邻接表深度优先遍历

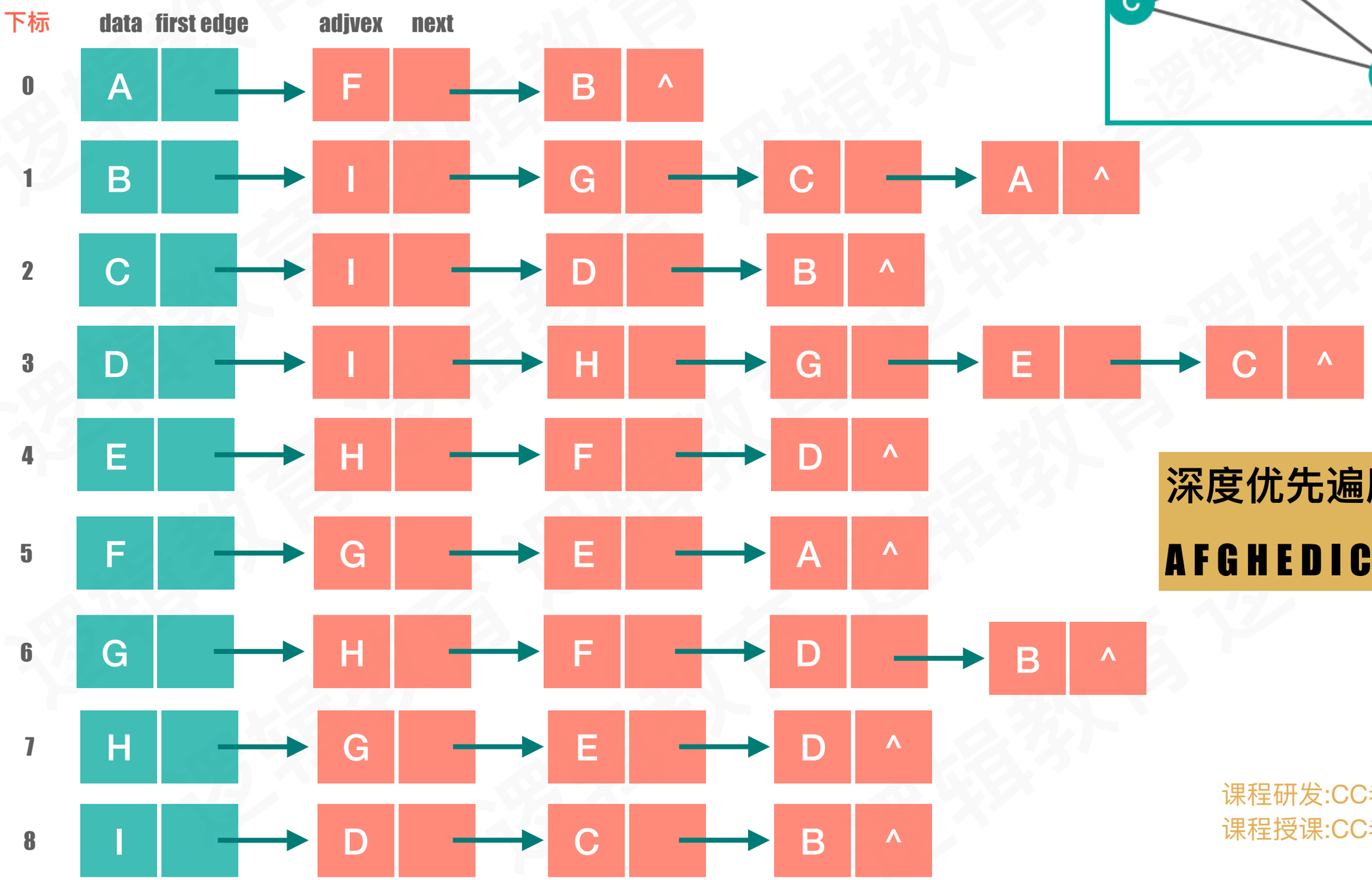




逻辑教育
Logic education

图的遍历—邻接表深度优先遍历





深度优先遍历结果:

AFGHEDICB

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历—邻接表深度优先遍历代码实现思路

1. 利用邻接矩阵将信息存储到邻接表中

2. 创建一个visited 数组,用来标识顶点是否已经被遍历过.

3. 初始化visited 数组,将数组中元素置为FALSE

4. 选择顶点开始遍历.(注意非连通图的情况)

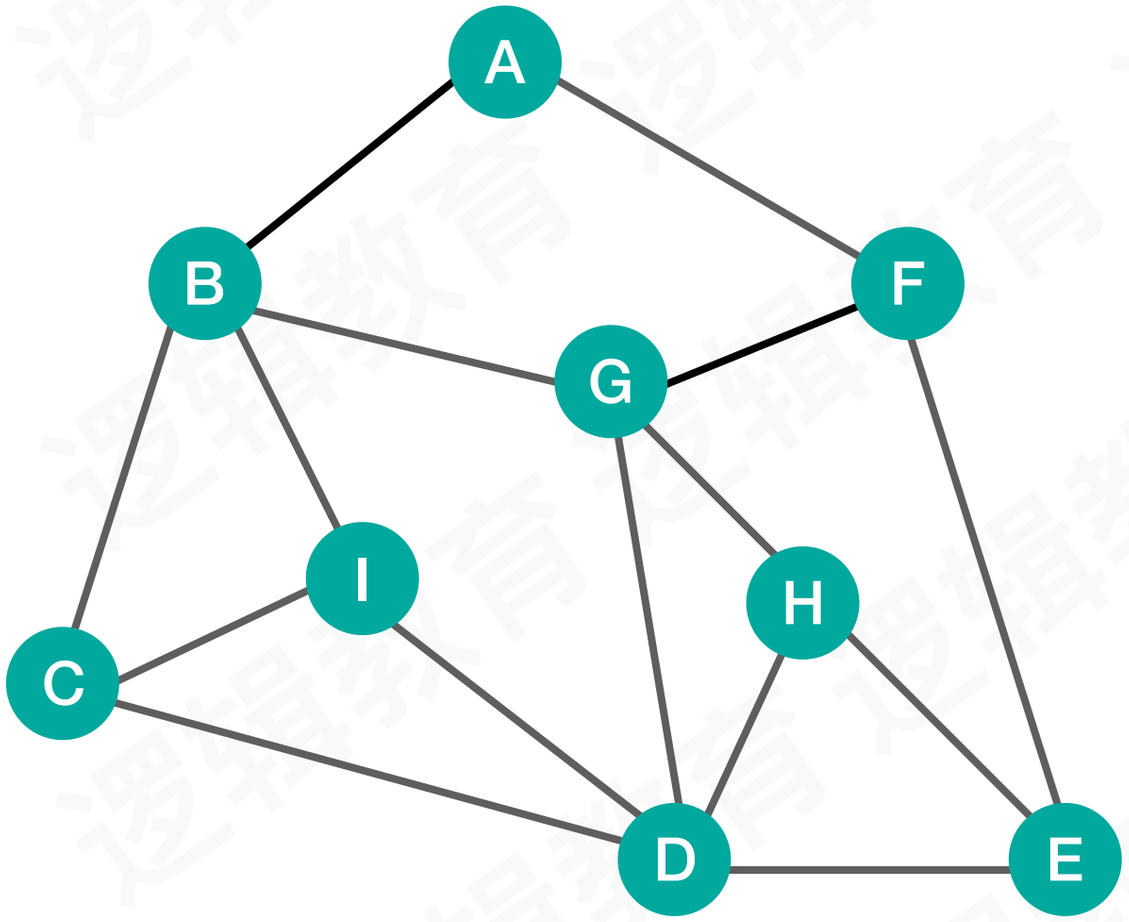
5. 进入递归; 打印i 对应的顶点信息. 并将该顶点标识为已遍历.

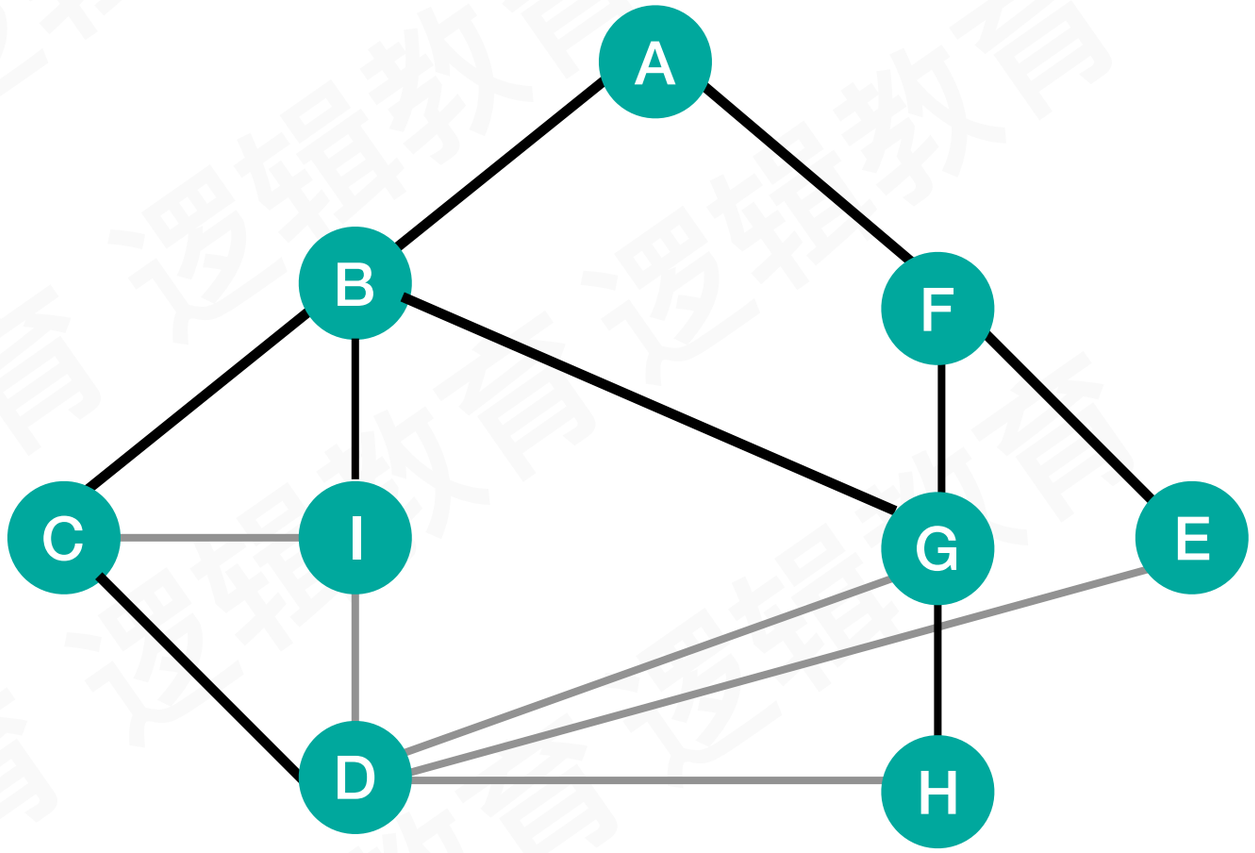
6. 循环遍历边表,判断当前顶点 是否等于1,并且当前该顶点没有被遍历过,则继续递归 DFS;

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

图的遍历—广度优先遍历





课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

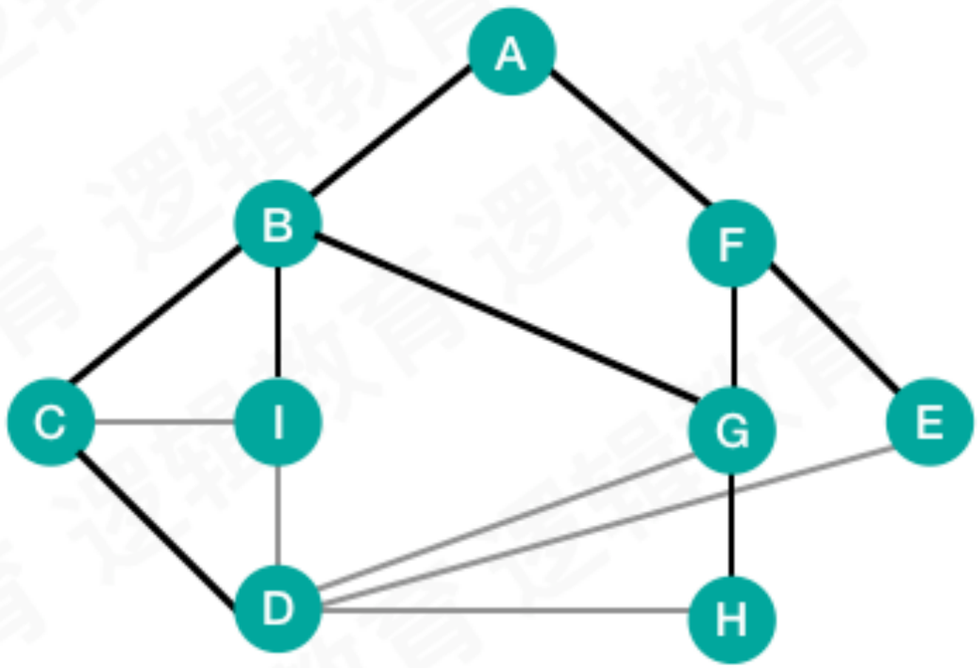
图的遍历—广度优先遍历特点

1、把根节点放到队列的末尾。
2、每次从队列的头部取出一个元素，查看这个元素所有的下一级元素，把它们放到队列的末尾。并把这个元素记为它下一级元素的前驱。
3、找到所要找的元素时结束程序。
4、如果遍历整个树还没有找到，结束程序.

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

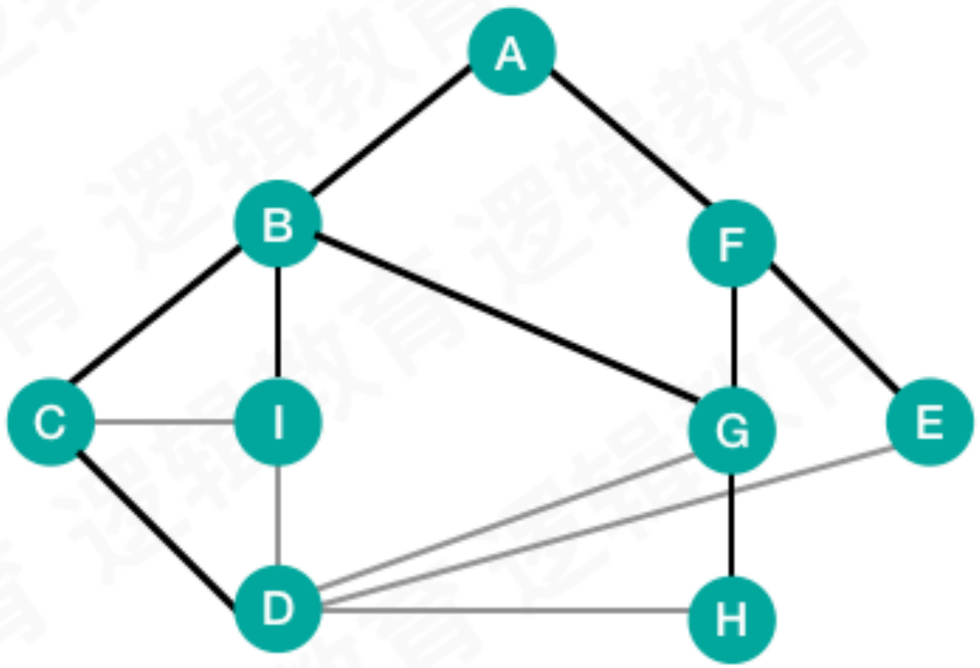
图的遍历—广度优先遍历



结果		出队列	入队列
		/	
/	①	A	
A	②	B F	
B	③	F C I G	
F	④	C I G E	
C	⑤	I G E D	
I	⑥	G E D	
G	⑦	E D H	
E	⑧	D H	
D	⑨	H	
H	⑩		

逻辑教育
Logic education

图的遍历—广度优先遍历



		出队列	入队列
/	①	A	
A	②	B F	
B	③	F C I G	
F	④	C I G E	
C	⑤	I G E D	
I	⑥	G E D	
G	⑦	E D H	
E	⑧	D H	
D	⑨	H	
H	⑩		

逻辑教育
Logic education

图的遍历—邻接矩阵/邻接表广度优先遍历

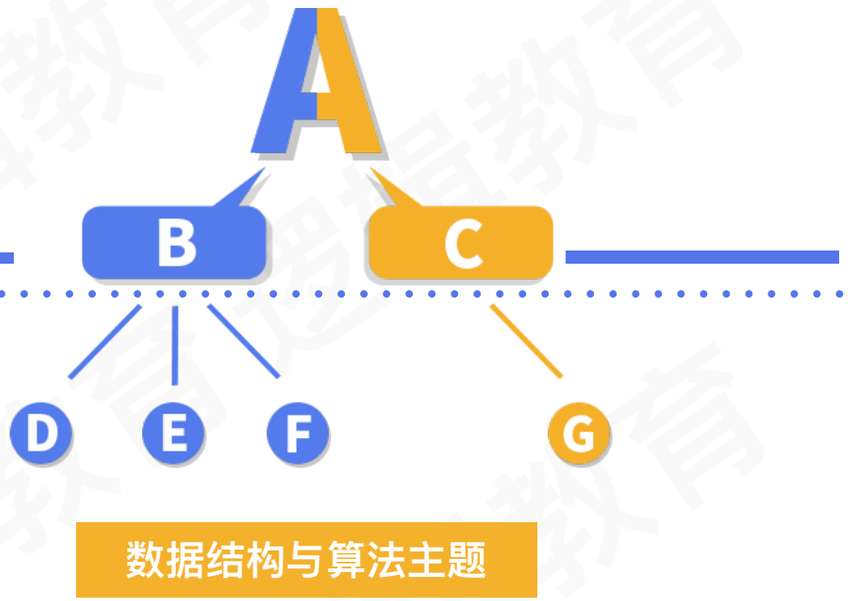
详细代码实现-请阅读胖C DEMO

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

Class Ending !

thanks, see you next time



@CC老师

全力以赴，非同凡"想"

课程研发:CC老师
课程授课:CC老师

逻辑教育
Logic education

课程研发:CC老师
课程授课:CC老师