

本节内容

图
基本操作

王道考研/CSKAOYAN.COM

1

知识总览

图的基本操作:

- $\text{Adjacent}(G,x,y)$: 判断图 G 是否存在边 $\langle x,y \rangle$ 或 (x,y) 。
- $\text{Neighbors}(G,x)$: 列出图 G 中与结点 x 邻接的边。
- $\text{InsertVertex}(G,x)$: 在图 G 中插入顶点 x 。
- $\text{DeleteVertex}(G,x)$: 从图 G 中删除顶点 x 。
- $\text{AddEdge}(G,x,y)$: 若无向边 (x,y) 或有向边 $\langle x,y \rangle$ 不存在, 则向图 G 中添加该边。
- $\text{RemoveEdge}(G,x,y)$: 若无向边 (x,y) 或有向边 $\langle x,y \rangle$ 存在, 则从图 G 中删除该边。
- $\text{FirstNeighbor}(G,x)$: 求图 G 中顶点 x 的第一个邻接点, 若有则返回顶点号。若 x 没有邻接点或图中不存在 x , 则返回-1。
- $\text{NextNeighbor}(G,x,y)$: 假设图 G 中顶点 y 是顶点 x 的一个邻接点, 返回除 y 之外顶点 x 的下一个邻接点的顶点号, 若 y 是 x 的最后一个邻接点, 则返回-1。
- $\text{Get_edge_value}(G,x,y)$: 获取图 G 中边 (x,y) 或 $\langle x,y \rangle$ 对应的权值。
- $\text{Set_edge_value}(G,x,y,v)$: 设置图 G 中边 (x,y) 或 $\langle x,y \rangle$ 对应的权值为 v 。

图的存储

邻接矩阵

邻接表

十字链表

邻接多重表

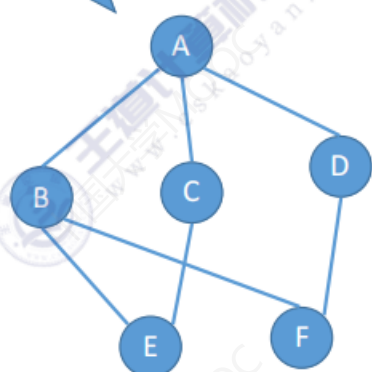
王道考研/CSKAOYAN.COM

2

图的基本操作

- Adjacent(G,x,y): 判断图G是否存在边<x, y>或(x, y)。

无向图



邻接矩阵

$O(1)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

$O(1) \sim O(|V|)$

	data	*first
0	A	1 → 2 → 3 ^
1	B	0 → 4 → 5 ^
2	C	0 → 4 ^
3	D	0 → 5 ^
4	E	1 → 2 ^
5	F	1 → 3 ^

王道考研/CSKAOYAN.COM

3

图的基本操作

- Adjacent(G,x,y): 判断图G是否存在边<x, y>或(x, y)。



邻接矩阵

$O(1)$

	data		A	B	C	D	E	F
0	A	A	0	1	0	0	0	0
1	B	B	0	0	0	0	0	0
2	C	C	1	0	0	0	0	0
3	D	D	1	0	0	0	0	0
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

$O(1) \sim O(|V|)$

	data	*first
0	A	1 ^
1	B	^
2	C	0 ^
3	D	0 ^
4	E	1 → 2 ^
5	F	1 → 3 ^

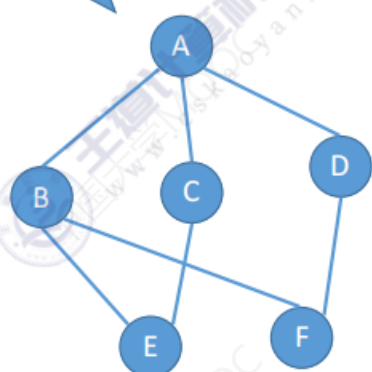
王道考研/CSKAOYAN.COM

4

图的基本操作

- Neighbors(G,x): 列出图G中与结点x邻接的边。

无向图



邻接矩阵

$O(|V|)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0



邻接表

$O(1) \sim O(|V|)$

	data	*first
0	A	1 → 2 → 3 ^
1	B	0 → 4 → 5 ^
2	C	0 → 4 ^
3	D	0 → 5 ^
4	E	1 → 2 ^
5	F	1 → 3 ^

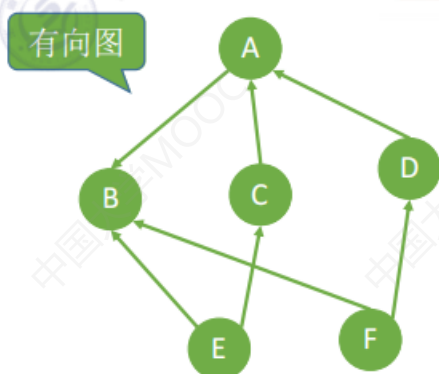
王道考研/CSKAOYAN.COM

5

图的基本操作

- Neighbors(G,x): 列出图G中与结点x邻接的边。

有向图



邻接矩阵

$O(|V|)$

	data		A	B	C	D	E	F
0	A	A	0	1	0	0	0	0
1	B	B	0	0	0	0	0	0
2	C	C	1	0	0	0	0	0
3	D	D	1	0	0	0	0	0
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

万一是个稀疏图呢?

邻接表

出边: $O(1) \sim O(|V|)$
入边: $O(|E|)$

	data	*first
0	A	1 ^
1	B	^
2	C	0 ^
3	D	0 ^
4	E	1 → 2 ^
5	F	1 → 3 ^

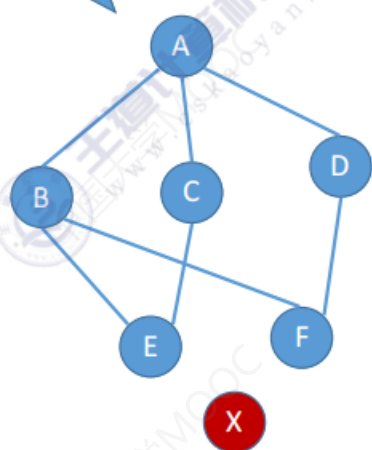
王道考研/CSKAOYAN.COM

6

图的基本操作

- InsertVertex(G,x): 在图G中插入顶点x。

无向图



邻接矩阵

$O(1)$

	data		A	B	C	D	E	F	X
0	A	A	0	1	1	1	0	0	0
1	B	B	1	0	0	0	1	1	0
2	C	C	1	0	0	0	1	0	0
3	D	D	1	0	0	0	0	1	0
4	E	E	0	1	1	0	0	0	0
5	F	F	0	1	0	1	0	0	0
6	X	X	0	0	0	0	0	0	0

邻接表

$O(1)$

	data	*first
0	A	1 → 2 → 3 ^
1	B	0 → 4 → 5 ^
2	C	0 → 4 ^
3	D	0 → 5 ^
4	E	1 → 2 ^
5	F	1 → 3 ^
6	X	^

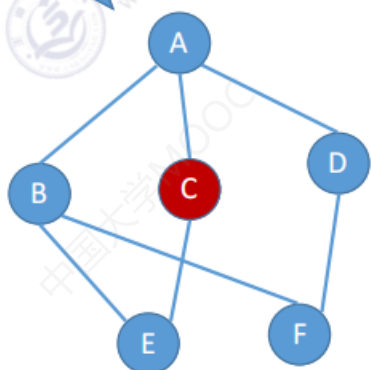
王道考研/CSKAOYAN.COM

7

图的基本操作

- DeleteVertex(G,x): 从图G中删除顶点x。

无向图



邻接矩阵

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

	data	*first
0	A	1 → 2 → 3 ^
1	B	0 → 4 → 5 ^
2	C	0 → 4 ^
3	D	0 → 5 ^
4	E	1 → 2 ^
5	F	1 → 3 ^

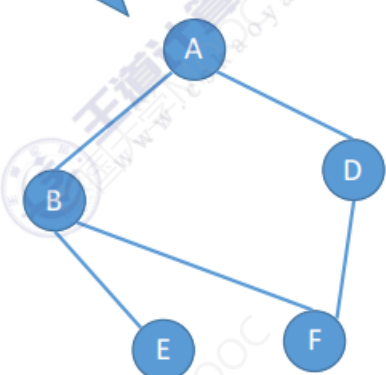
王道考研/CSKAOYAN.COM

8

图的基本操作

- DeleteVertex(G,x): 从图G中删除顶点x。

无向图



邻接矩阵

$O(|V|)$

	data		A	B	空	D	E	F
0	A	A	0	1	0	1	0	0
1	B	B	1	0	0	0	1	1
2	空	空	0	0	0	0	0	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	0	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

$O(1) \sim O(|E|)$

data	*first
0	A → 1 → 3 ^
1	B → 0 → 4 → 5 ^
2	空 ^
3	D → 0 → 5 ^
4	E → 1 ^
5	F → 1 → 3 ^

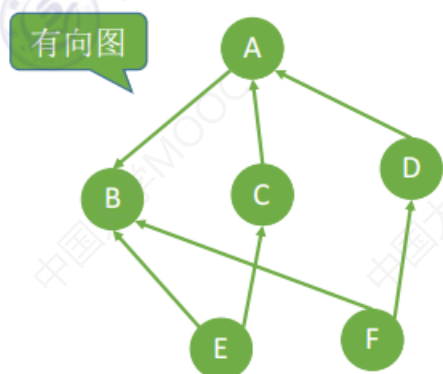
王道考研/CSKAOYAN.COM

9

图的基本操作

- DeleteVertex(G,x): 从图G中删除顶点x。

有向图



邻接矩阵

$O(|V|)$

	data		A	B	C	D	E	F
0	A	A	0	1	0	0	0	0
1	B	B	0	0	0	0	0	0
2	C	C	1	0	0	0	0	0
3	D	D	1	0	0	0	0	0
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

删出边: $O(1) \sim O(|V|)$
删入边: $O(|E|)$

data	*first
0	A → 1 ^
1	B ^
2	C → 0 ^
3	D → 0 ^
4	E → 1 → 2 ^
5	F → 1 → 3 ^

王道考研/CSKAOYAN.COM

10

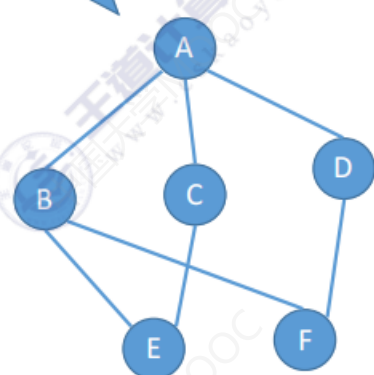
图的基本操作

- **AddEdge(G,x,y)**: 若无向边(x, y)或有向边<x, y>不存在, 则向图G中添加该边。

懒!

有向图也类似

无向图



邻接矩阵 $O(1)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表 $O(1)$

	data	*first
0	A	→ 1 → 2 → 3 ^
1	B	→ 0 → 4 → 5 ^
2	C	→ 0 → 4 ^
3	D	→ 0 → 5 ^
4	E	→ 1 → 2 ^
5	F	→ 1 → 3 ^

王道考研/CSKAOYAN.COM

11

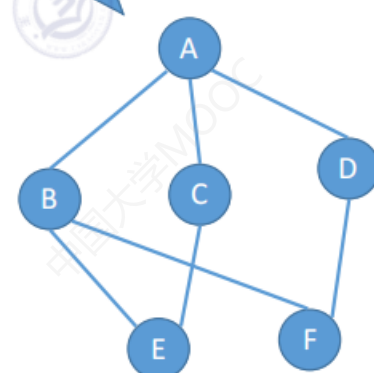
图的基本操作

- **RemoveEdge(G,x,y)**: 若无向边(x, y)或有向边<x, y>存在, 则从图G中删除该边。

懒!

有向图也类似

无向图



邻接矩阵 $O(1)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表 $O(1) \sim O(|V|)$

	data	*first
0	A	→ 1 → 2 → 3 ^
1	B	→ 0 → 4 → 5 ^
2	C	→ 0 → 4 ^
3	D	→ 0 → 5 ^
4	E	→ 1 → 2 ^
5	F	→ 1 → 3 ^

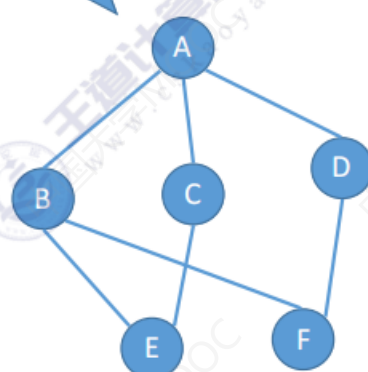
王道考研/CSKAOYAN.COM

12

图的基本操作

- **FirstNeighbor(G,x)**: 求图G中顶点x的第一个邻接点, 若有则返回顶点号。若x没有邻接点或图中不存在x, 则返回-1。

无向图



邻接矩阵

$O(1) \sim O(|V|)$

data		A	B	C	D	E	F
0	A	A	0	1	1	0	0
1	B	B	1	0	0	1	1
2	C	C	1	0	0	0	1
3	D	D	1	0	0	0	1
4	E	E	0	1	1	0	0
5	F	F	0	1	0	1	0

邻接表

$O(1)$

data	*first
0	A → 1 → 2 → 3 → ^
1	B → 0 → 4 → 5 → ^
2	C → 0 → 4 → ^
3	D → 0 → 5 → ^
4	E → 1 → 2 → ^
5	F → 1 → 3 → ^

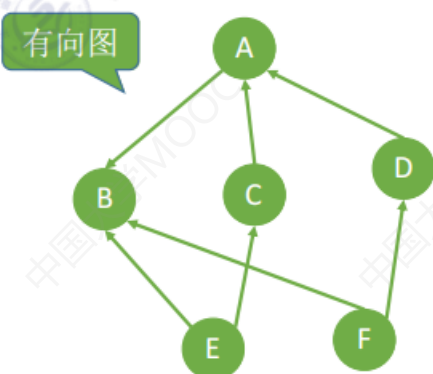
王道考研/CSKAOYAN.COM

13

图的基本操作

- **FirstNeighbor(G,x)**: 求图G中顶点x的第一个邻接点, 若有则返回顶点号。若x没有邻接点或图中不存在x, 则返回-1。

有向图



邻接矩阵

$O(1) \sim O(|V|)$

data		A	B	C	D	E	F
0	A	A	0	1	0	0	0
1	B	B	0	0	0	0	0
2	C	C	1	0	0	0	0
3	D	D	1	0	0	0	0
4	E	E	0	1	1	0	0
5	F	F	0	1	0	1	0

邻接表

找出边邻接点: $O(1)$

找入边邻接点: $O(1) \sim O(|E|)$

data	*first
0	A → 1 → ^
1	B → ^
2	C → 0 → ^
3	D → 0 → ^
4	E → 1 → 2 → ^
5	F → 1 → 3 → ^

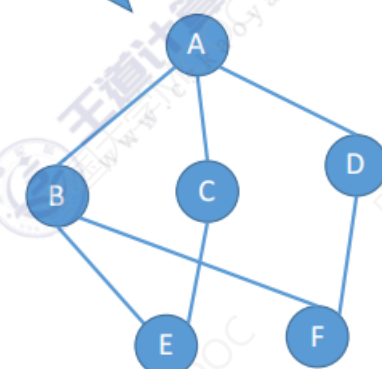
王道考研/CSKAOYAN.COM

14

图的基本操作

- **NextNeighbor(G,x,y):** 假设图G中顶点y是顶点x的一个邻接点, 返回除y之外顶点x的下一个邻接点的顶点号, 若y是x的最后一个邻接点, 则返回-1。

无向图



邻接矩阵

$O(1) \sim O(|V|)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

$O(1)$

	data	*first
0	A	→ 1 → 2 → 3 → ^
1	B	→ 0 → 4 → 5 → ^
2	C	→ 0 → 4 → ^
3	D	→ 0 → 5 → ^
4	E	→ 1 → 2 → ^
5	F	→ 1 → 3 → ^

王道考研/CSKAOYAN.COM

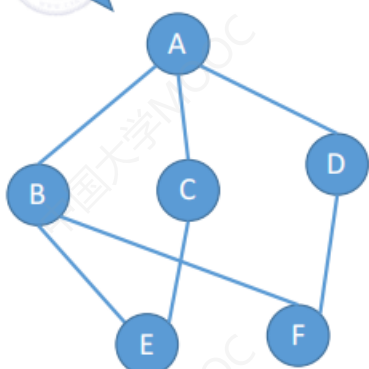
15

图的基本操作

- **Get_edge_value(G,x,y):** 获取图G中边(x,y)或<x,y>对应的权值。
- **Set_edge_value(G,x,y,v):** 设置图G中边(x,y)或<x,y>对应的权值为v。
- **Adjacent(G,x,y):** 判断图G是否存在边<x,y>或(x,y)。

雷同, 核心在于找到边

无向图



邻接矩阵

$O(1)$

	data		A	B	C	D	E	F
0	A	A	0	1	1	1	0	0
1	B	B	1	0	0	0	1	1
2	C	C	1	0	0	0	1	0
3	D	D	1	0	0	0	0	1
4	E	E	0	1	1	0	0	0
5	F	F	0	1	0	1	0	0

邻接表

$O(1) \sim O(|V|)$

	data	*first
0	A	→ 1 → 2 → 3 → ^
1	B	→ 0 → 4 → 5 → ^
2	C	→ 0 → 4 → ^
3	D	→ 0 → 5 → ^
4	E	→ 1 → 2 → ^
5	F	→ 1 → 3 → ^

王道考研/CSKAOYAN.COM

16

知识回顾与重要考点

- $\text{Adjacent}(G, x, y)$: 判断图 G 是否存在边 $\langle x, y \rangle$ 或 (x, y) 。
- $\text{Neighbors}(G, x)$: 列出图 G 中与结点 x 邻接的边。
- $\text{InsertVertex}(G, x)$: 在图 G 中插入顶点 x 。
- $\text{DeleteVertex}(G, x)$: 从图 G 中删除顶点 x 。
- $\text{AddEdge}(G, x, y)$: 若无向边 (x, y) 或有向边 $\langle x, y \rangle$ 不存在, 则向图 G 中添加该边。
- $\text{RemoveEdge}(G, x, y)$: 若无向边 (x, y) 或有向边 $\langle x, y \rangle$ 存在, 则从图 G 中删除该边。
- $\text{FirstNeighbor}(G, x)$: 求图 G 中顶点 x 的第一个邻接点, 若有则返回顶点号。若 x 没有邻接点或图中不存在 x , 则返回-1。
- $\text{NextNeighbor}(G, x, y)$: 假设图 G 中顶点 y 是顶点 x 的一个邻接点, 返回除 y 之外顶点 x 的下一个邻接点的顶点号, 若 y 是 x 的最后一个邻接点, 则返回-1。
- $\text{Get_edge_value}(G, x, y)$: 获取图 G 中边 (x, y) 或 $\langle x, y \rangle$ 对应的权值。
- $\text{Set_edge_value}(G, x, y, v)$: 设置图 G 中边 (x, y) 或 $\langle x, y \rangle$ 对应的权值为 v 。

此外, 还有图的遍历算法, 包括深度优先遍历和广度优先遍历。

王道考研/CSKAOYAN.COM

17



@王道论坛



@王道计算机考研备考

@王道咸鱼老师-计算机考研

@王道楼楼老师-计算机考研



等撩



@王道计算机考研



等撩

知乎

@王道计算机考研

微信视频号

@王道计算机考研

微信公众平台

@王道在线

18