

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program

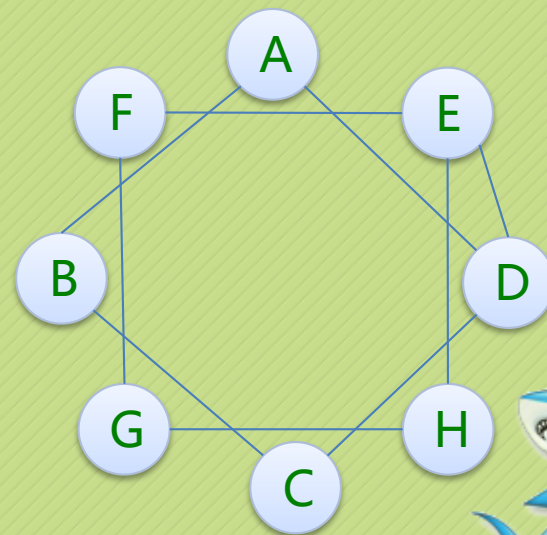
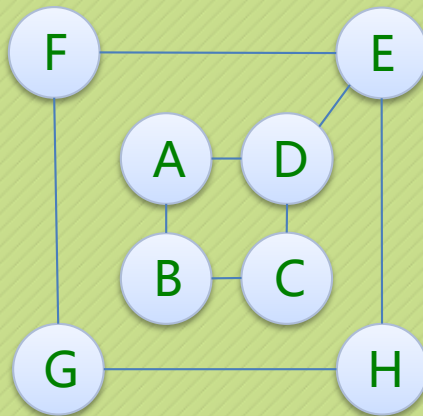
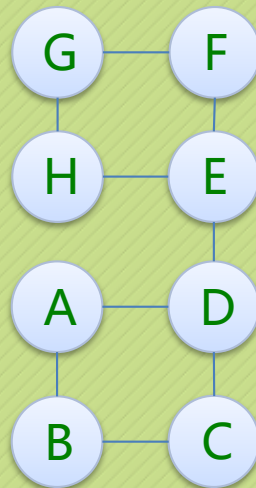
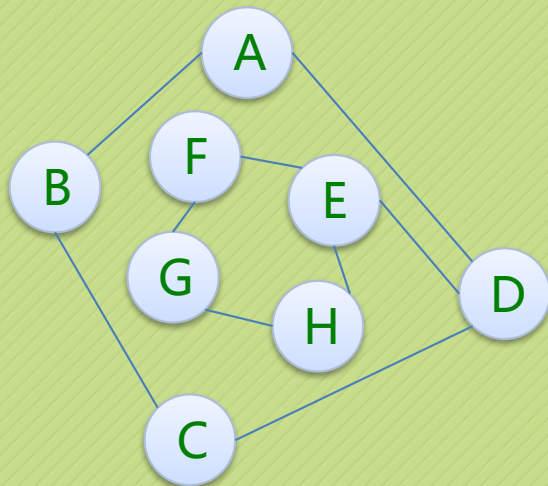


图的存储结构

- 图的存储结构相比较线性表与树来说就复杂很多。
- 我们回顾下，对于线性表来说，是一一对一的关系，所以用数组或者链表均可简单存放。树结构是一对多的关系，所以我们要将数组和链表的特性结合在一起才能更好的存放。
- 那么我们的图，是多对多的情况，另外图上的任何一个顶点都可以被看作是第一个顶点，任一顶点的邻接点之间也不存在次序关系。
- 我们仔细观察以下几张图，然后深刻领悟一下：

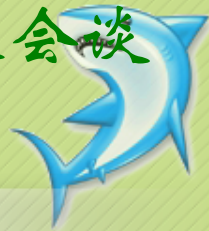


图的存储结构



图的存储结构

- 因为任意两个顶点之间都可能存在联系，因此无法以数据元素在内存中的物理位置来表示元素之间的关系（内存物理位置是线性的，图的元素关系是平面的）。
- 如果用多重链表来描述倒是可以做到，但在几节课前的树章节我们已经讨论过，纯粹用多重链表导致的浪费是无法想像的（如果各个顶点的度数相差太大，就会造成巨大的浪费）。
- 所幸，前辈们已经帮想好了出路，我们接下来会谈图的五种不同的存储结构，大家做好准备哦~



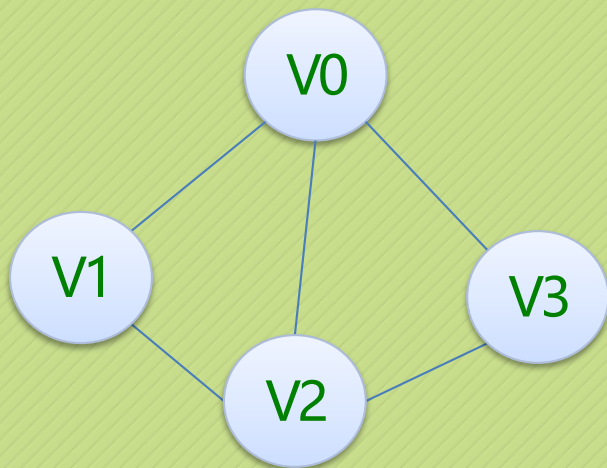
邻接矩阵 (无向图)

- 考虑到图是由顶点和边或弧两部分组成，合在一起比较困难，那就很自然地考虑到分为两个结构来分别存储。
- 顶点因为不区分大小、主次，所以用一个一维数组来存储是很不错的选择。
- 而边或弧由于是顶点与顶点之间的关系，一维数组肯定就搞不定了，那我们不妨考虑用一个二维数组来存储。
- 于是我们的邻接矩阵方案就诞生了！



邻接矩阵 (无向图)

- 图的邻接矩阵 (Adjacency Matrix) 存储方式是用两个数组来表示图。一个一维数组存储图中顶点信息，一个二维数组 (称为邻接矩阵) 存储图中的边或弧的信息。



顶点数组:	V0	V1	V2	V3
	V0	V1	V2	V3
V0	0	1	1	1
V1	1	0	1	0
V2	1	1	0	1
V3	1	0	1	0



邻接矩阵 (无向图)

- 我们可以设置两个数组，顶点数组为 $\text{vertex}[4]=\{V0,V1,V2,V3\}$ ，边数组 $\text{arc}[4][4]$ 为对称矩阵(0表示不存在顶点间的边，1表示顶点间存在边)。
- 对称矩阵：所谓对称矩阵就是 n 阶矩阵的元满足 $a[i][j]=a[j][i](0 \leq i,j \leq n)$ 。即从矩阵的左上角到右下角的主对角线为轴，右上角的元与左下角相对应的元全都是相等的。



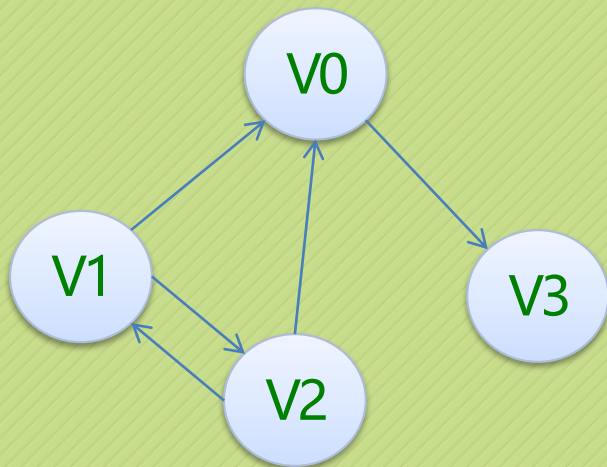
邻接矩阵 (无向图)

- 有了这个二维数组组成的对称矩阵, 我们就可以很容易地知道图中的信息:
 - 要判定任意两顶点是否有边无边就非常容易了;
 - 要知道某个顶点的度, 其实就是这个顶点 V_i 在邻接矩阵中第 i 行(或第 i 列)的元素之和;
 - 求顶点 V_i 的所有邻接点就是将矩阵中第 i 行元素扫描一遍, $\text{arc}[i][j]$ 为1就是邻接点咯。



邻接矩阵 (有向图)

- 无向图的边构成了一个对称矩阵，貌似浪费了一半的空间，那如果是有向图来存放，会不会把资源都利用得很好呢？



顶点数组:	V0	V1	V2	V3
	V0	V1	V2	V3
V0	0	0	0	1
V1	1	0	1	0
V2	1	1	0	0
V3	0	0	0	0



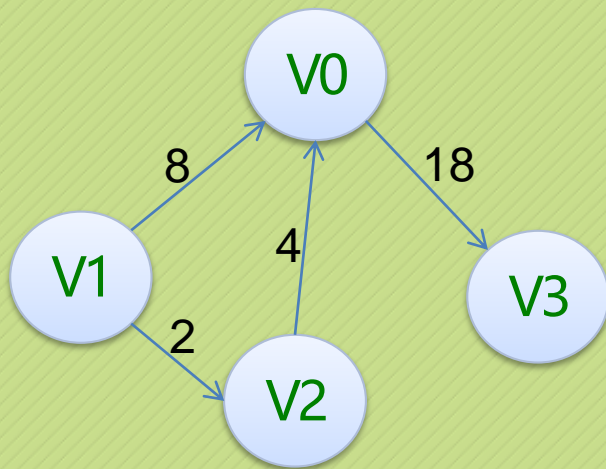
邻接矩阵 (有向图)

- 可见顶点数组 $\text{vertex}[4]=\{V0,V1,V2,V3\}$, 弧数组 $\text{arc}[4][4]$ 也是一个矩阵, 但因为是有向图, 所以这个矩阵并不对称, 例如由 $V1$ 到 $V0$ 有弧, 得到 $\text{arc}[1][0]=1$, 而 $V0$ 到 $V1$ 没有弧, 因此 $\text{arc}[0][1]=0$ 。
- 另外有向图是有讲究的, 要考虑入度和出度, 顶点 $V1$ 的入度为 1, 正好是第 $V1$ 列的各数之和, 顶点 $V1$ 的出度为 2, 正好是第 $V1$ 行的各数之和。



邻接矩阵 (网)

- 在图的术语中，我们提到了网这个概念，事实上也就是每条边上带有权的图就叫网。



顶点数组:	V0	V1	V2	V3
	V0	V1	V2	V3
V0	0	∞	∞	18
V1	8	0	2	∞
V2	4	∞	0	∞
V3	∞	∞	∞	0

- 这里“ ∞ ”表示一个计算机允许的、大于所有边上权值的值。



代码实现

- 作为一个课后作业给大家自己锻炼下，小甲鱼提供的参考答案仅供参考借鉴！

