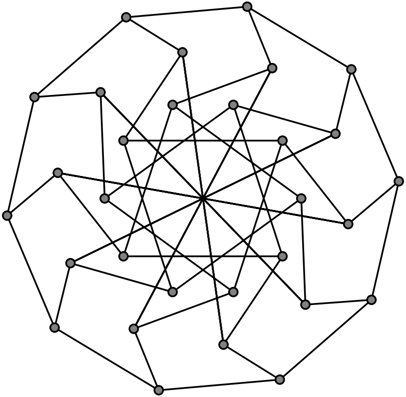
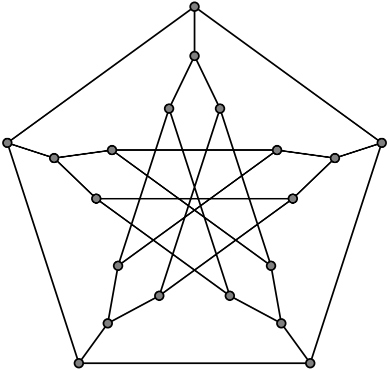
数据结构
Data Structure

Xia Tian
Email: xiat(at)ruc.edu.cn

Renmin University of China

Graph





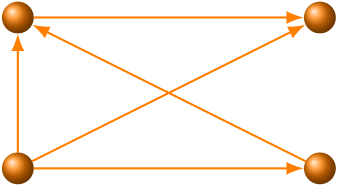
Content

- 图的定义
- 图的存储表示
- 图的遍历
- 图的连通性

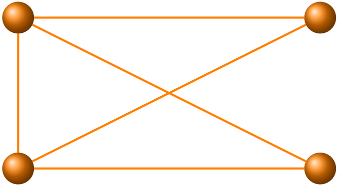
图 (Graph)

- 图 $G = (V, E)$, V 是顶点 (Vertex) 集合,E 是边/弧 (Edge/Arc) 的集合.
- 顶点的度、出度和入度

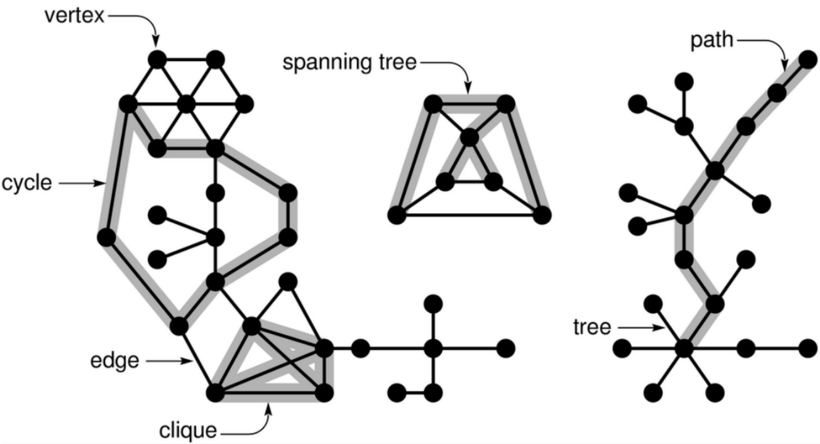
有向图:



无向图:



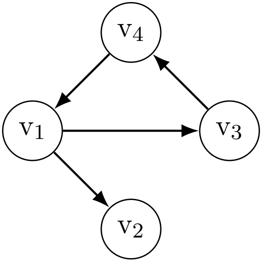
图的相关概念



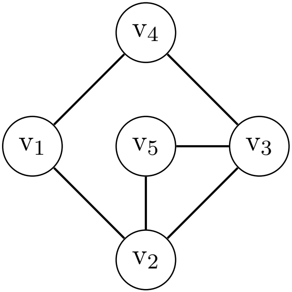
图的存储

如何表达下图的信息?

有向图:



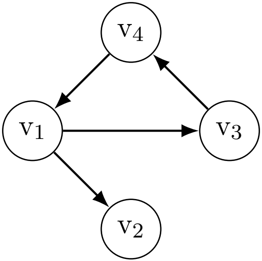
无向图:



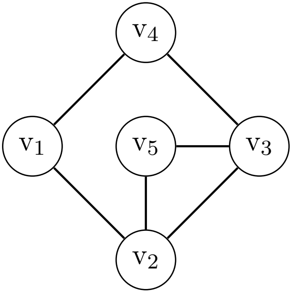
图的存储

如何表达下图的信息?

有向图:

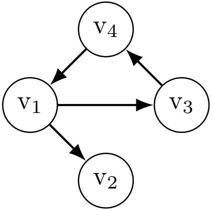


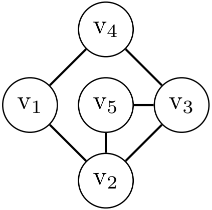
无向图:



- 可用邻接矩阵表达顶点及其关系。

图的存储



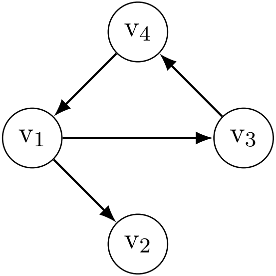


$$
\begin{array}{c c}
 & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

$$
\begin{array}{c c}
 & \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}
\end{array}
$$

- 根据邻接矩阵, 如何判断各顶点的度?

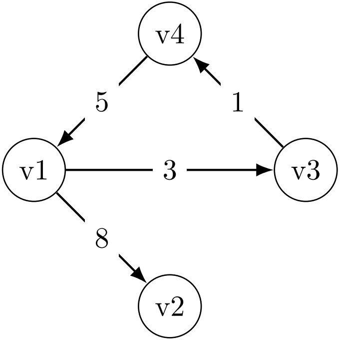
有向图的连续存储方式：邻接矩阵



- 建立二维数组 $A[n][n],\ n = |V|$
- 另需存放 n 个顶点信息

$$
\begin{array}{cc}
 & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}
\end{array}
$$

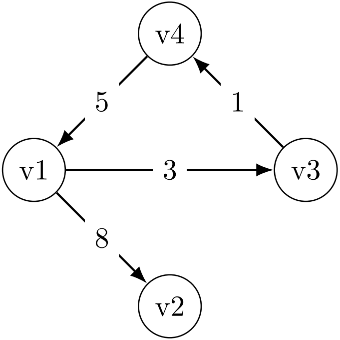
网的邻接矩阵



$$
\begin{array}{c|cccc}
 & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & \infty & 8 & 3 & \infty \\
v_2 & \infty & \infty & \infty & \infty \\
v_3 & \infty & \infty & \infty & 1 \\
v_4 & 5 & \infty & \infty & \infty
\end{array}
$$

- 有些图的边带有权重 (常用来表示成本、距离、时间等), 这样的图称为: 网。
- 网的邻接矩阵表达权重, 没有边的顶点之间的权重默认为 ∞
- 邻接矩阵表示方法非常直观、简单, 但是会有什么问题?

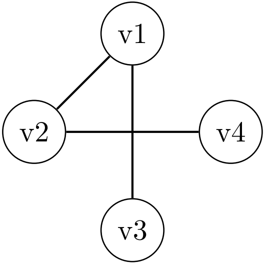
网的邻接矩阵

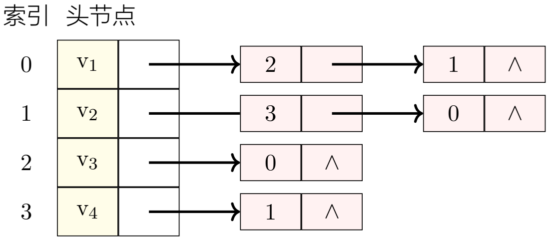


$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4
\end{array}
\begin{array}{c}
\begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\
\begin{pmatrix}
\infty & 8 & 3 & \infty \\
\infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 1 \\
5 & \infty & \infty & \infty
\end{pmatrix}
\end{array}
$$

- 有些图的边带有权重 (常用来表示成本、距离、时间等), 这样的图称为: 网。
- 网的邻接矩阵表达权重, 没有边的顶点之间的权重默认为 ∞
- 邻接矩阵表示方法非常直观、简单, 但是会有什么问题?
- 现实中的图经常对应稀疏矩阵, 在这样情形下会有很大空间浪费.

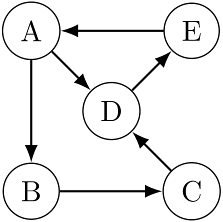
邻接表 (Adjacency List) – 无向图





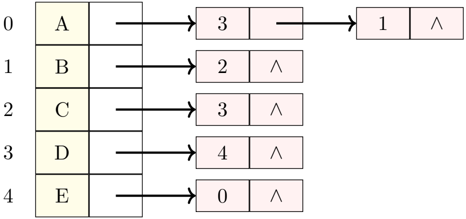
- 无向图的邻接表: 同一个顶点发出的边链接在同一个边链表中, 便于确定顶点的度
- 需要 n 个头结点, 2e 个表结点

邻接表–有向图

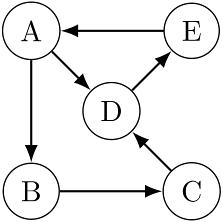


邻接表, 便于确定节点出度

索引　头节点

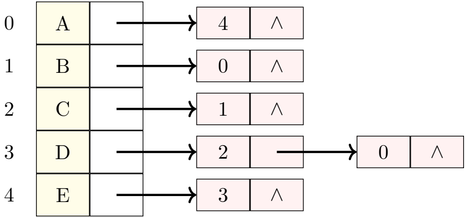


邻接表–有向图



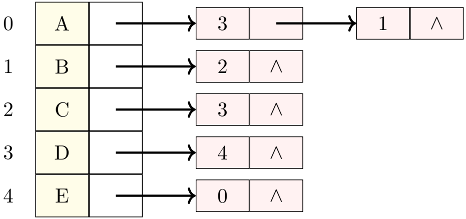
逆邻接表，便于确定节点入度

索引　头节点

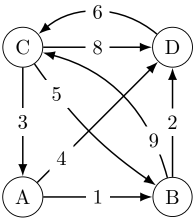


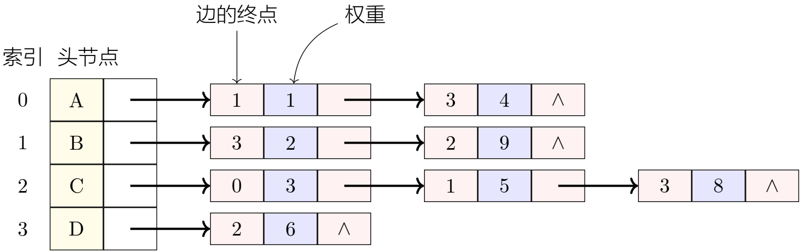
邻接表，便于确定节点出度

索引　头节点



邻接表–权重处理





练习

1. 请写出数组存储和邻接表的类型定义
2. 请在如下方面对比数组表示法和邻接表示法
   - 存储表示是否唯一
   - 空间复杂度
   - 操作 a: 求顶点 v_i 的度
   - 操作 b: 判定 (v_i, v_j) 是否是图的一条边
   - 操作 c: 通过遍历求边的数目

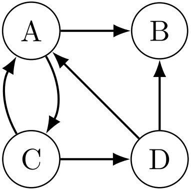
比较

	数组表示法	邻接表法
表示结果	唯一	不唯一
空间复杂度	$O(n^2)$ (适用于稠密图)	$O(n+e)$ (适用于稀疏图)
无向图求顶点 v_i 的度	第 i 行 (或第 i 列) 上非零元素的个数	第 i 个边表中的结点个数
有向图求顶点 v_i 的度	第 i 行上非零元素的个数是 v_i 出度, 第 i 列上非零元素的个数是 v_i 的入度	第 i 个边表上的结点个数, 求入度还需遍历各顶点的边表。逆邻接表则相反
判定 (v_i, v_j) 是否是图的一条边	看矩阵中的 i 行 j 列是否为 0	扫描第 i 个边表
求边的数目	检测整个矩阵中的非零元所耗费的时间是 $O(N^2)$	对每个边表的结点个数计数所耗费的时间是 $O(e+n)$

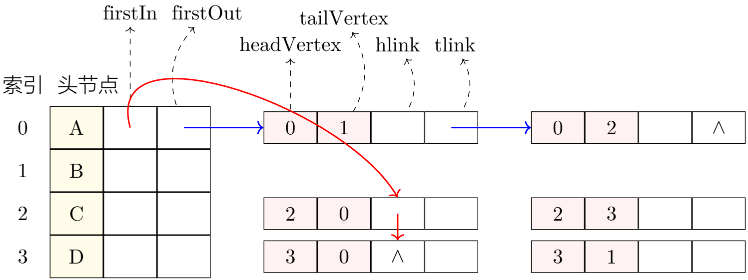
思考

怎么把邻接表和逆邻接表相结合，同时表示出来?

有向图的十字链表 (Orthogonal List)



将邻接表、逆邻接表结合起来.



有向图的十字链表

```
class VertexNode {
  String data;
  ArcBox firstIn;
  ArcBox firstOut;
}

class ArcBox {
  int headVertex, tailVertex;
  ArcBox hlink;
  ArcBox tlink;
  String data;
}
```

```
class OLGraph {
  List<VertexNode> xlist;
  int vertexNum, arcNum;
}
```

无向图的多重邻接表