

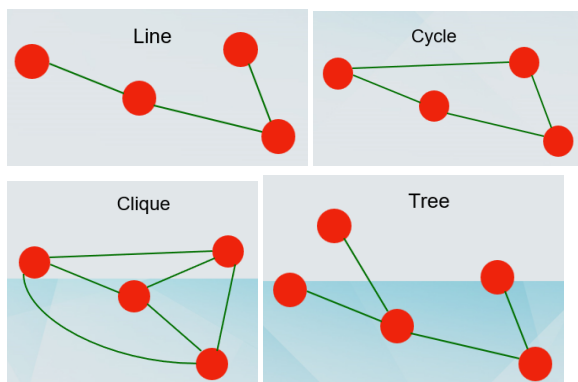
03.1 - Graph Definitions

图的符号与定义

对于图 $G = (V, E)$

- V 是顶点集合, $|V| = n$
- E 是边集合, $|E| = m$
- 无向边: $e = \{v, w\}$, $v, w \in V$
- 有向边: $e = (v, w)$, $v, w \in V$
- 顶点的邻居: 与顶点通过边相连的顶点的集合 (不包含次级顶点)
- 度: 顶点的邻居数量, 用 $\deg(v)$ 表示
 - 入度: 指向顶点的边的数量
 - 出度: 从顶点指向其他顶点的边的数量
- 路径: 一系列不重复的通过边相连的连续的顶点
 - 长度: 路径中的边数 (顶点数量 -1)
- 距离: 两个顶点之间最短路径的长度
- 直径 (diameter): 图中最长的最短路径的长度

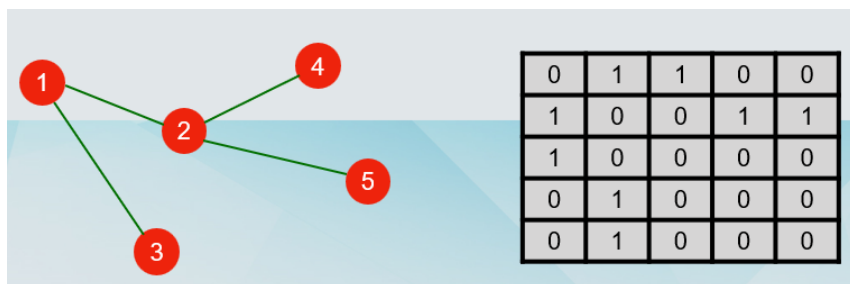
Line, Cycles, Trees, Cliques



图的表示 - 邻接矩阵与邻接表

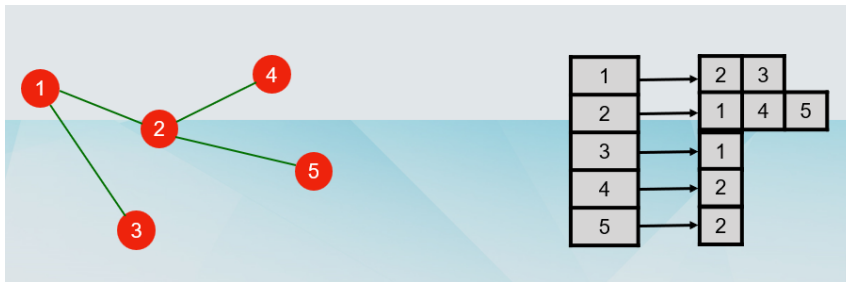
邻接矩阵 (Adjacency Matrix)

- 第 i 个节点对应第 i 行第 i 列
- 如果 i 、 j 节点之间有一条边, 那么 $A_{ij} = 1$; 如果没有, $A_{ij} = 0$ 。对于有权重的图, A_{ij} 就等于权重
- 对于无向图, $A_{ij} = A_{ji}$; 而对于有向图, 此条不成立

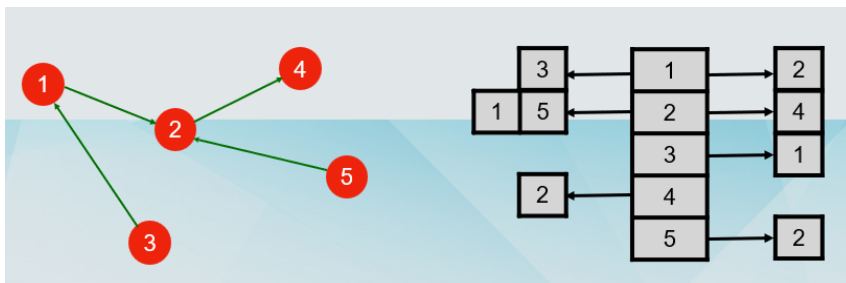


邻接表 (Adjacency List)

- 在邻接表中，节点被排列成一个列表，每个节点都指向它的邻居节点
- 对于无向图，节点只指向一个方向。节点 v 的邻接表包含所有与 v 直接相连的节点



- 对于有向图，节点指向两个方向，分别表示入度和出度
 - 入度：节点的邻列表包含所有指向该节点的边
 - 出度：节点的邻列表包含所有从该节点出发指向其他节点的边



比较

邻接矩阵

- $O(n^2)$ 空间： $n \times n$ 的矩阵
- 检查两个顶点 u 、 v 是否相邻： $O(1)$ 时间
 - 直接访问元素即可
- 找到顶点的所有邻接顶点： $O(n)$ 时间
 - 遍历顶点所在整行

邻接表

- $O(n + m)$ 空间：与顶点相连的顶点数量其实就是边的数量
- 检查两个顶点 u 、 v 是否相邻： $O(\min(\deg(u), \deg(v)))$
 - 需要遍历邻接表中一个顶点的邻居列表来检查是否存在边
- 找到顶点的所有邻接顶点： $O(\deg u)$
 - 只需遍历顶点 u 的邻接表

在稀疏图（边的数量接近于顶点数量）邻接表比邻接矩阵更合适，内存使用更为高效，但检查两个顶点是否相邻的时间复杂度较高

而对于邻接矩阵，其适用于边较多的稠密图，因为可以在常数时间内检查两个顶点是否相邻，但内存消耗较大