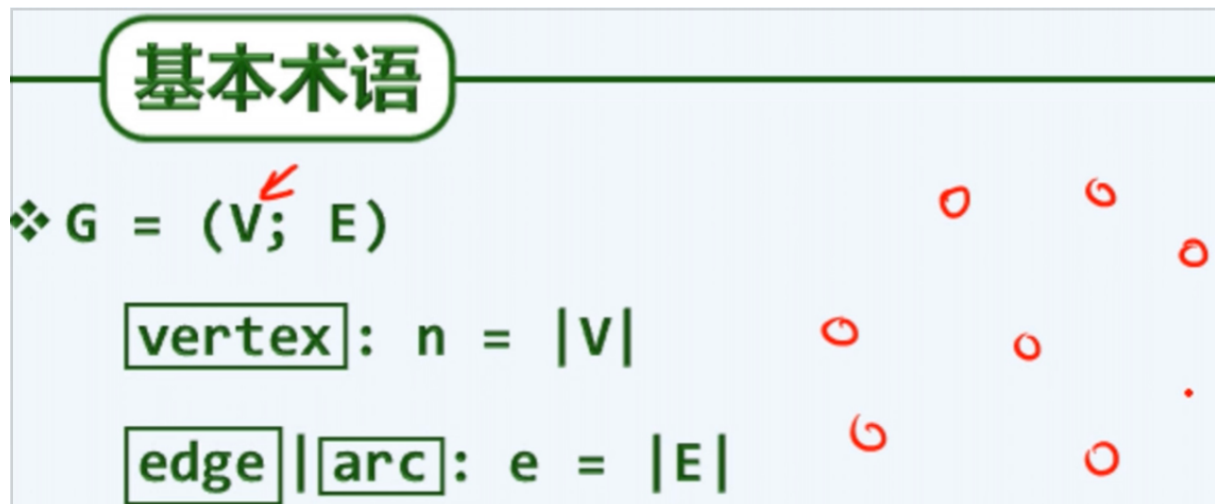


06 图 | 06A 概述

#数据结构邓神

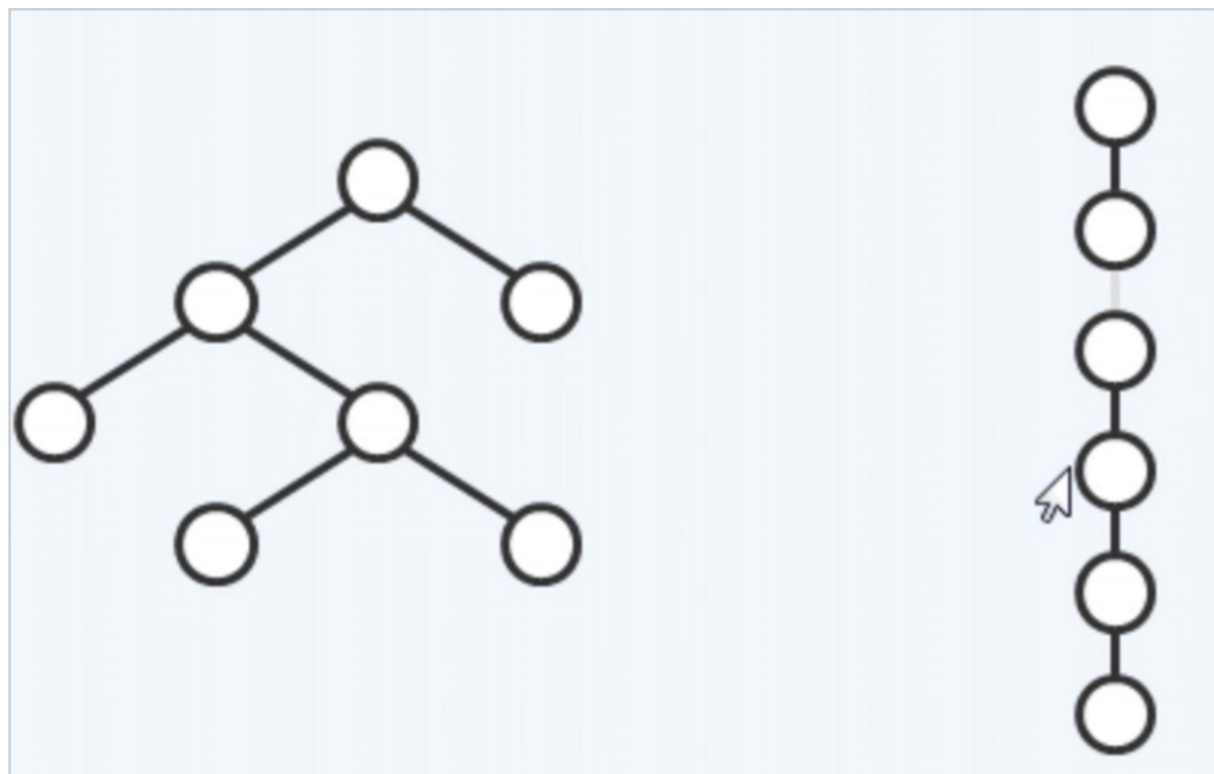
- 术语
- 实现
- 算法



Adjacency 邻接（顶点与顶点） $V \sim E$

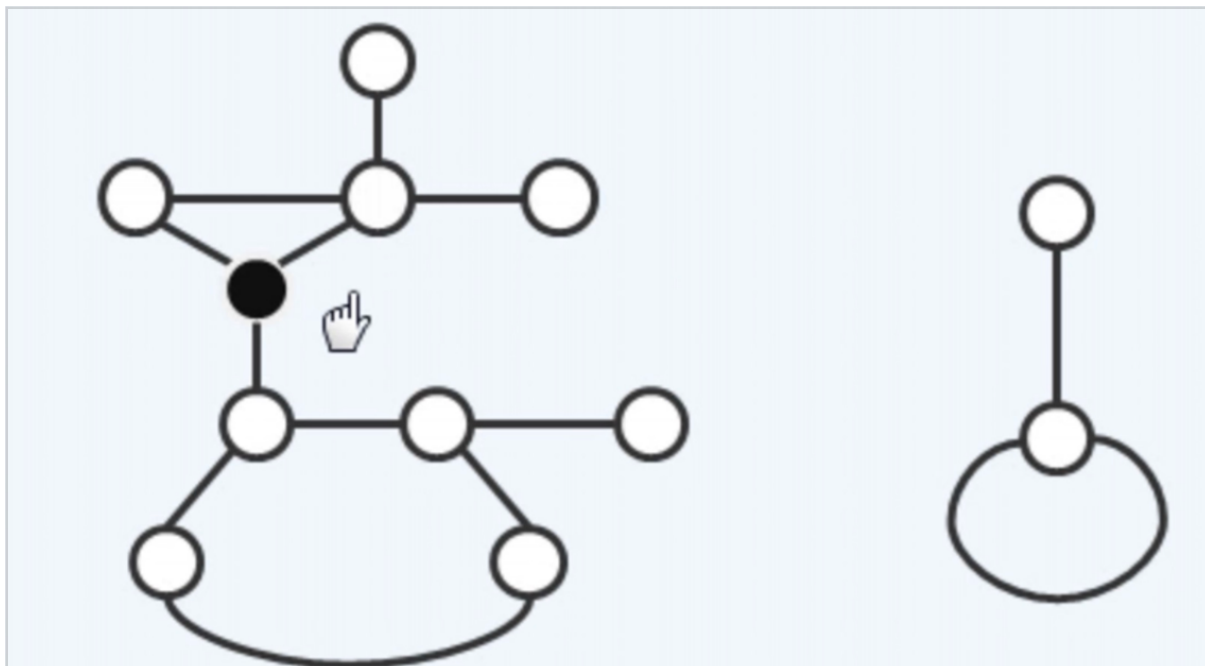
Incidence 关联（顶点与边的关系） $V \sim E$

我们可以把我们之前所学的几种数据结构视为图的特例



只能在唯一前驱和唯一后继定义邻接关系

图：



在计算机科学中，我们不允许节点连接自己

有向图 | 无向图

无向图/有向图

❖ 若邻接顶点 u 和 v 的次序无所谓

则 (u, v) 为无向边 **undirected edge**

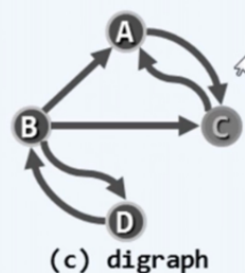
例 若 u 是 v 的好友，则 v 也必是 u 的好友

- 如果一幅图中所有的边都是无向的，就是无向图

❖ 反之，有向图 **digraph** 中均为有向边 **directed edge**

u 、 v 分别称作边 (u, v) 的尾 (tail)、头 (head)

例 “ u 欠 v 的钱”与“ v 欠 u 的钱”



- 如果一幅图中所有的边都是有向的，就是有向图

- 如果一幅图有些是有方向的有些是没有方向的就叫做混合图 mixed

我们主要是研究有向图，因为任何一个无向图和混合图转换为有向图

用无向图为例，我们不妨认为无向的那一条边等价于有向图的彼此两条互指的边

路径 | 环路

路径/环路

❖ 路径 $\pi = \langle v_0, v_1, \dots, v_k \rangle$

长度 $|\pi| = k$

- 如果在一条通路中不含重复的节点较为 simple path
- 如包含重复的节点 就是 (non-simple) path

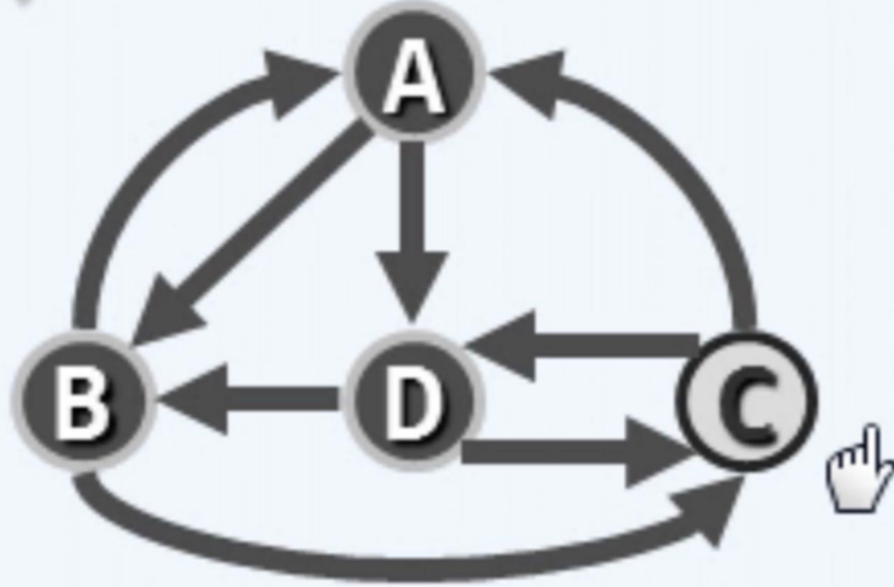
环路：起点 == 终点

一样的如果 环路中不包含重复的节点，就是简单环路

如果一个图中不包含任何的环路就称为有向无环图(DAG)

在欧拉环路中我们可以从任何一个点出发经过所有的边一次切构成一个环路

(i) Eulerian tour



哈密顿环路就是经过所有的顶点恰好一次的环路

(ii) Hamiltonian tour

