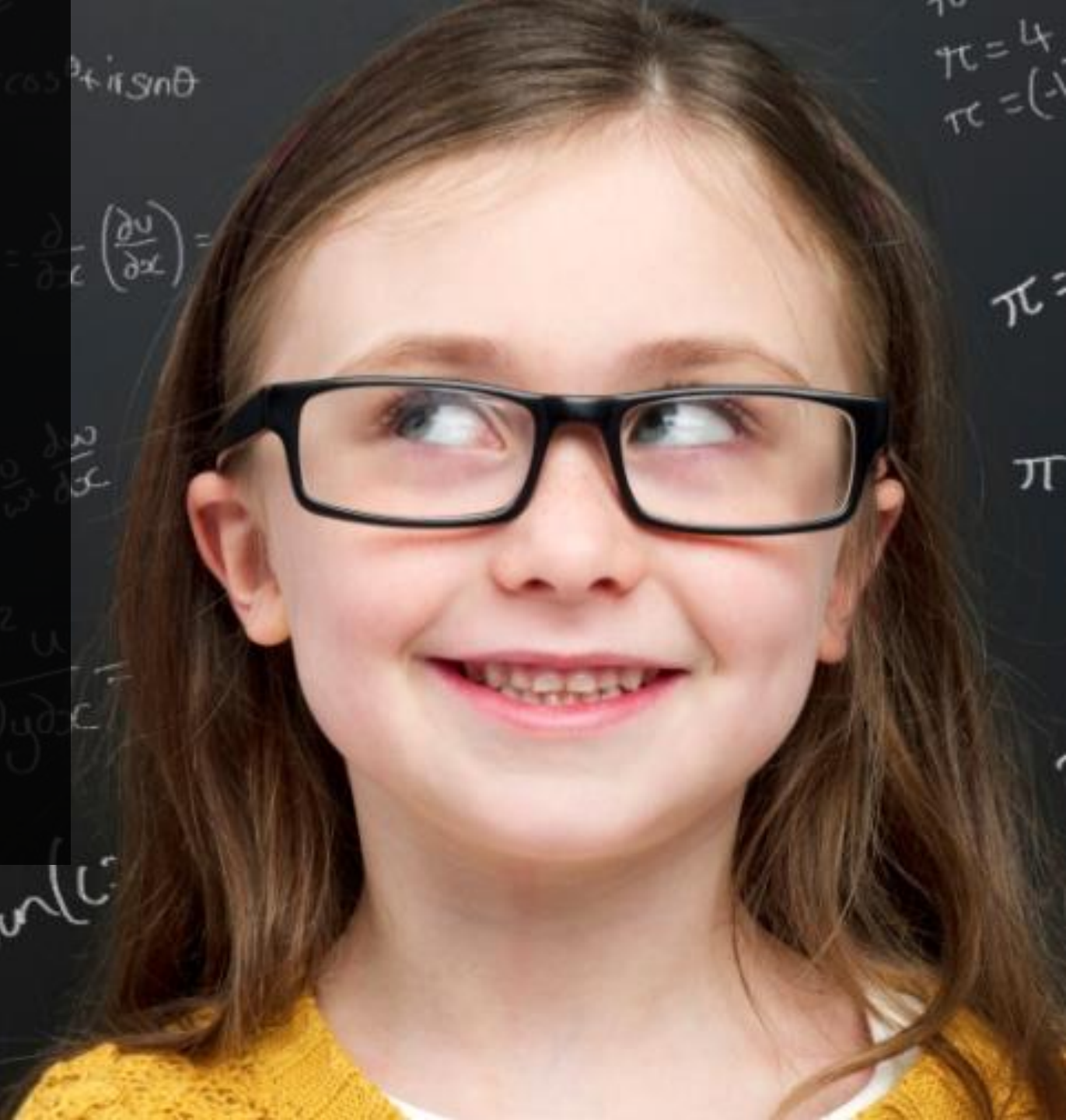


亲子算法课 (2)

——数据结构

作者：叶蒙蒙



数字化的世界

- 数字化的世界
- 计算机处理的都是数字
- 如果计算机要处理一个人或者一件事——用一个或者多个属性来表示这个人或事
- 属性在计算机中用数字表示，例如：
男：1
女：2



把数字组织起来

- 整齐的图书馆 vs 混乱的小房间
在哪里更容易找到需要的书？
- 把数字都收集到一起、放整齐



对数字进行计算 和操作

- 计算
 - 加减乘除.....
 - 大于、小于、等于.....
 - 复杂的函数
- 操作
 - 查找
 - 排序
 -



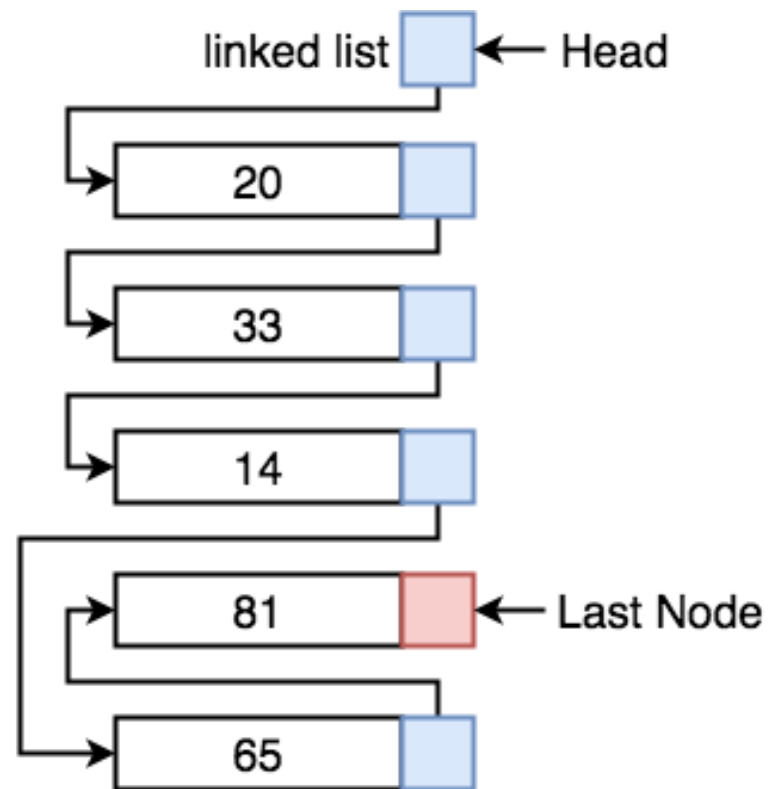
什么是数据结构 (Data Structure)?

- 计算机中存储、组织数据的方式
- 常见数据结构
 - 数组 (Array)
 - 链表 (Linked List)
 - 树 (Tree)
 - 图 (Graph)
 - 堆 (Heap)
 - 散列表 (Hash)
 -



	arr	
arr[0]	20	0x100
arr[1]	33	0x104
arr[2]	14	0x108
arr[3]	65	0x112
arr[4]	81	0x116

Array representation



数组和链表

数组和链表——相似

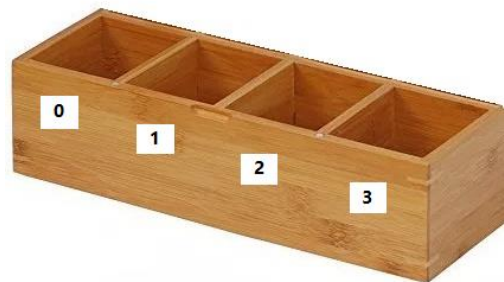


- 数字从前到后一个挨着一个
- 排成一队
- 除了首尾，每一个元素（图中的水果）都有且仅有两个“邻居”
- 每个元素的“地位”都是平等的





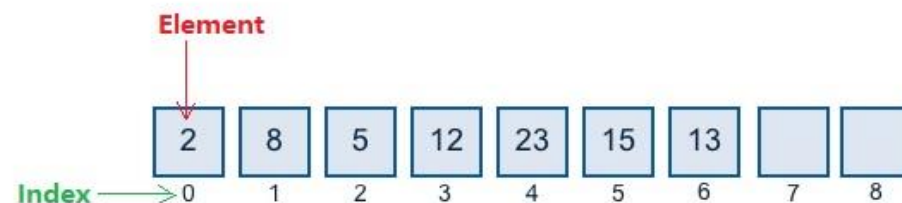
数组和链表—— 不同



- 数组：一排连在一起的“盒子”
- 链表：一列“火车”

数组（Array）：一排连在一起的“盒子”

- 盒子的个数在创建的时候确定，位置也在创建时固定——盒子之间的相互位置不会改变
- 盒子上有标号——根据盒子上的标号（index, 索引、下标）直接找到某一个盒子
- 每个盒子里面可以装东西（元素），也可以是空的
- 空着的盒子可以把东西放进去，有东西的盒子可以把东西拿出来
- 如果要把一个盒子里原有的东西换成新的，需要
 - i) 把原有的拿出来
 - ii) 把新的放进去



思考题

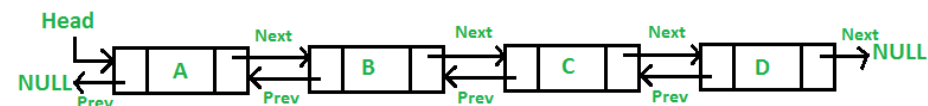
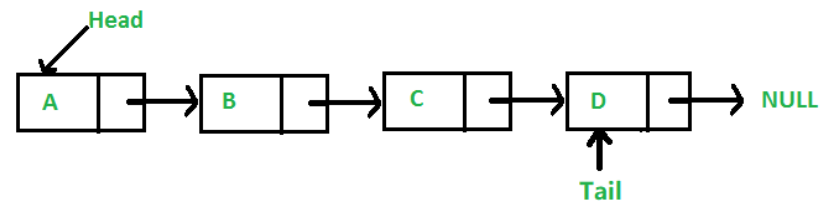


如果一个数组已经放满了

- 再想放新的元素进去，该怎么办？
- 想把里面的元素重新排列，该怎么办？

链表 (Linked List) —— 一列火车

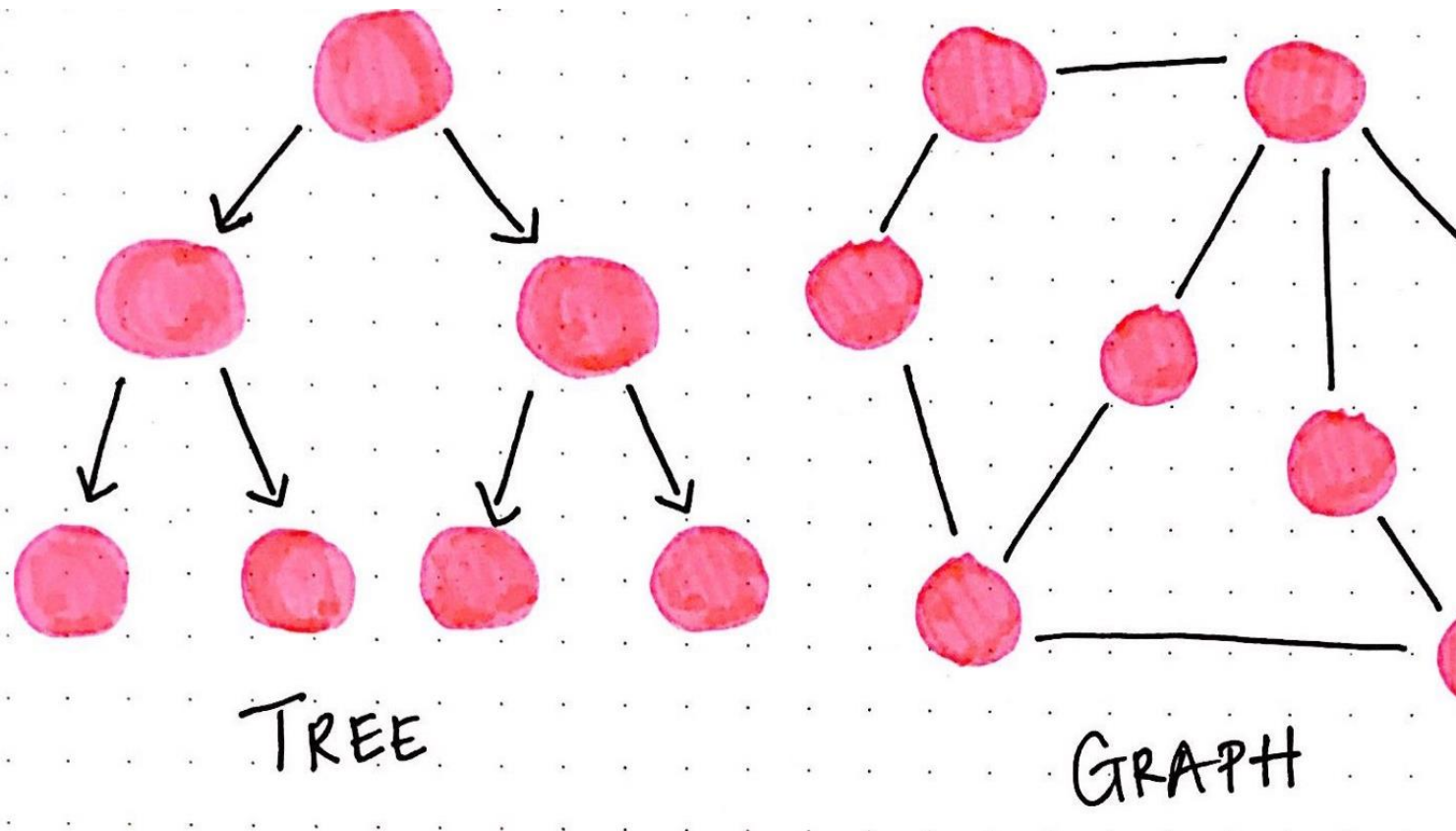
- 有车头和车尾——头尾和中间车厢不同
- 单向链表——只能从车头往后找车厢
- 双向链表——既可以从车头向后，也可以从车尾向前
- 车厢
 - 没有标号——找其中一列车厢，必须从车头（或者车尾）开始，不能用标号直接找到
 - 原来车厢可以“卸掉”；新的车厢可以加上——车厢个数可以改变；车厢之间的相互位置可以改变
- 车厢里一般都会有东西



思考题

- 和数组比，链表有什么优点？有什么缺点？
- 数组和链表，谁更“占地儿”？谁更方便“拿东西”？谁更方便“放东西”？

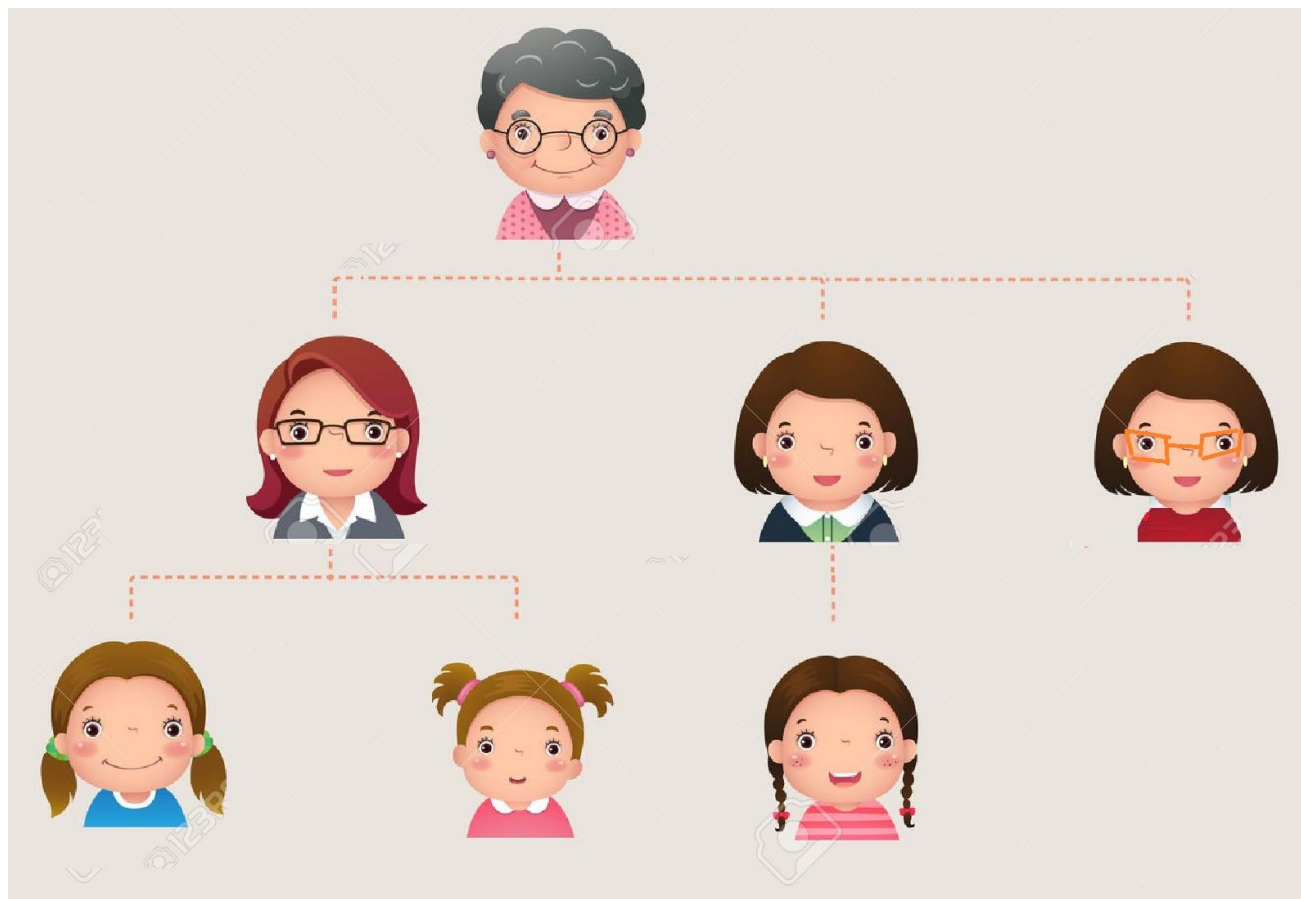


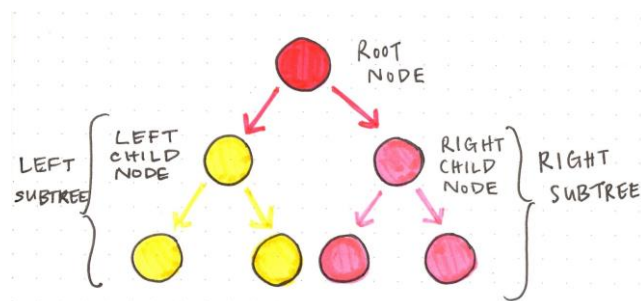
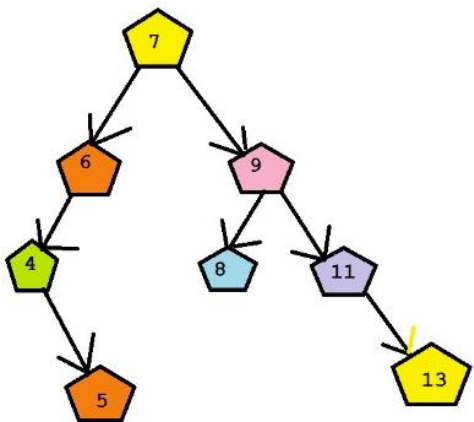


树和图

树 (Tree)

- 有很多层
- 每一层的节点地位不一样，相邻两层中
 - 上层是父母 (Parent)
 - 下层是子女 (Children)
- 有根，有枝，有叶
 - 一棵树只有一个根
 - 没有Children的元素是叶子
- 同一层的节点之间不相连，上下两层之间相连
- 一个节点
 - 向上只和一个Parent节点相连
 - 向下可以和0个，1个，2个或者更多个Children节点相连
- 树的一部分，还是树





二叉树

- 一棵树
- 每个节点最多有两个Children节点
- Children分为左右
 - 左子节点/左子树
 - 右子节点/右子树

思考:

- 如果每个节点最多有一个Children节点会怎么样?
- 如果每个节点最多有3, 4, 5, ...个节点会怎么样?

对二叉树的操作

(节点可以增加, 也可以减少)

- 构建
- 遍历!!!
 - 先序 (先根序)
 - 中序 (中根序)
 - 后序 (后根序)
- 查找
- 旋转.....

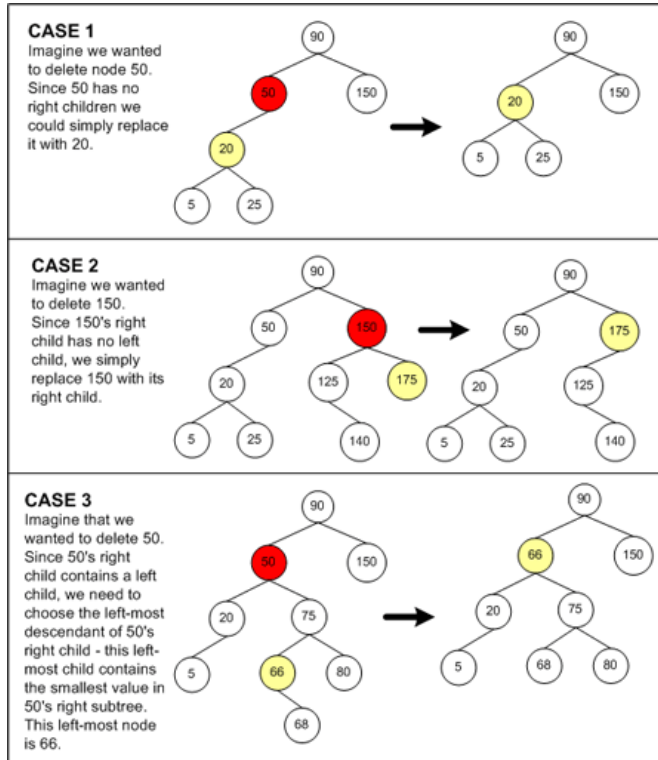
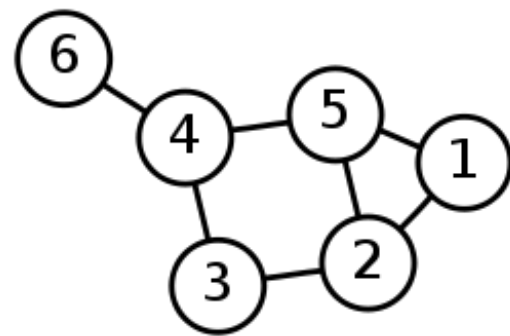
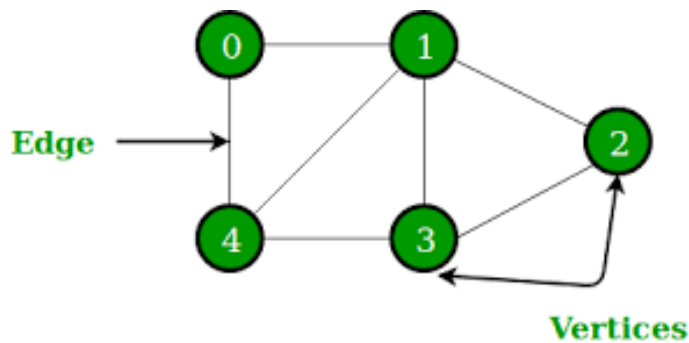
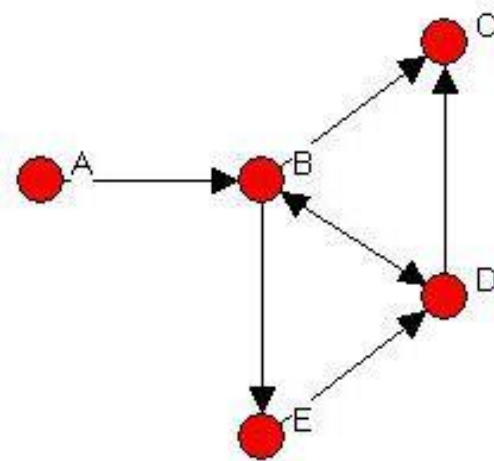
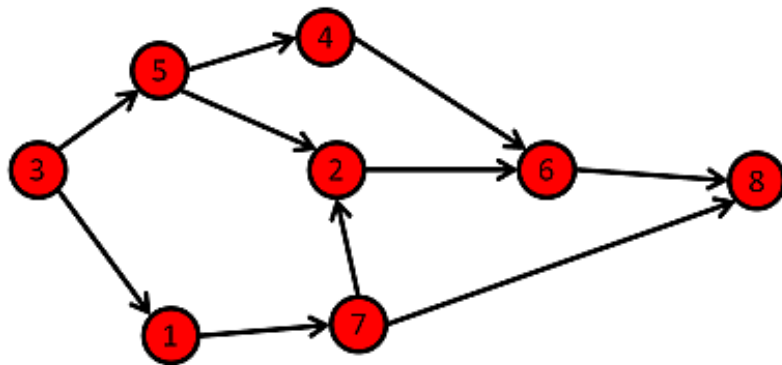


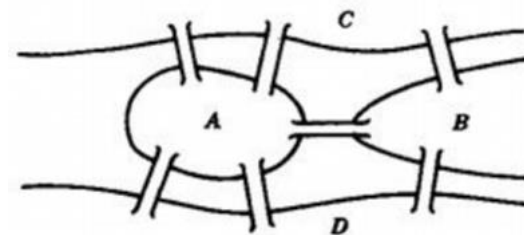
图 (Graph)

- 两个要素
 - 顶点 (Vertex)
 - 边 (Edge)
- 有 (方) 向 vs 无 (方) 向
 - 边才有可能有方向, 顶点和方向无关
- 有环 vs 无环



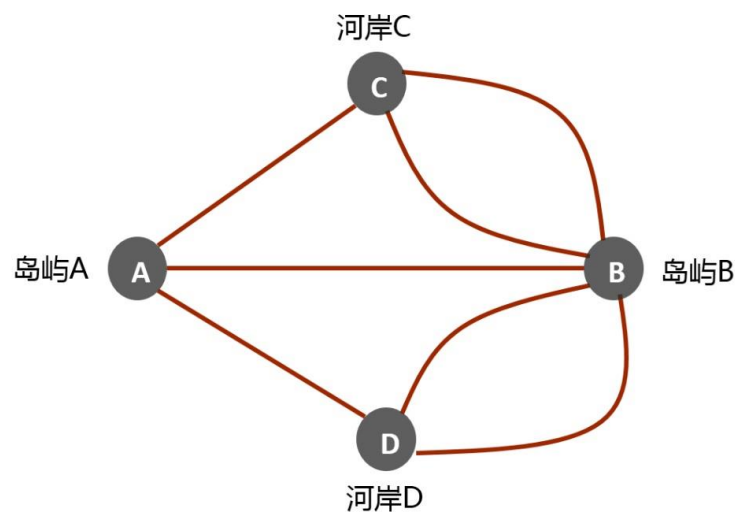
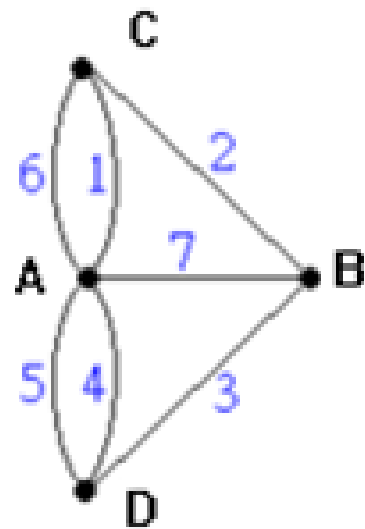
额外阅读：哥尼斯堡七桥问题

- 1736年，欧拉，普鲁士古城哥尼斯堡
- 普瑞格尔河正好从市中心流过，河中心有两座小岛，岛和两岸之间建筑有七座古桥。
- 试图每座桥恰好走过一遍并回到原出发点，但从来没人成功过

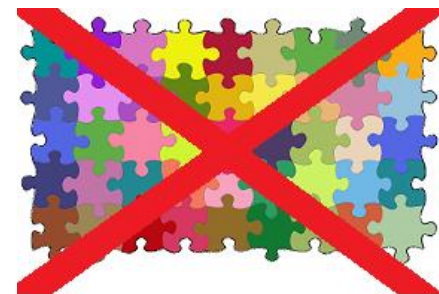


哥尼斯堡七桥问题

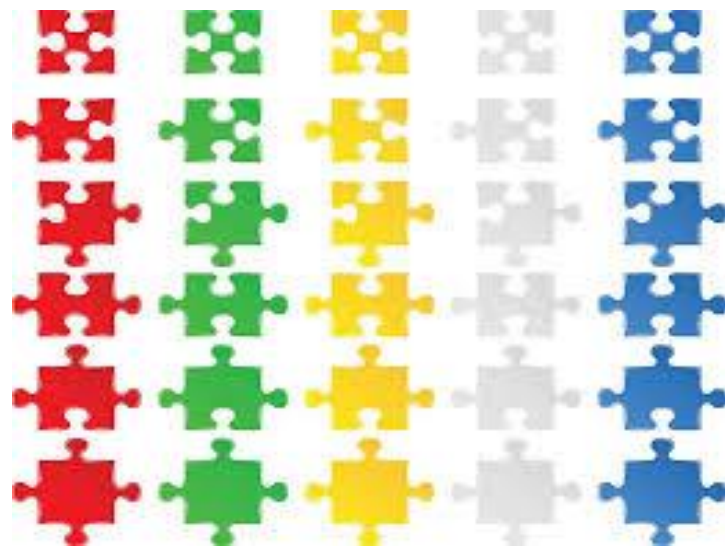
- 欧拉证明了：每座桥恰好走过一遍并回到原出发点——这种走法是不可能的



不同的数据结构之间
不能混用！

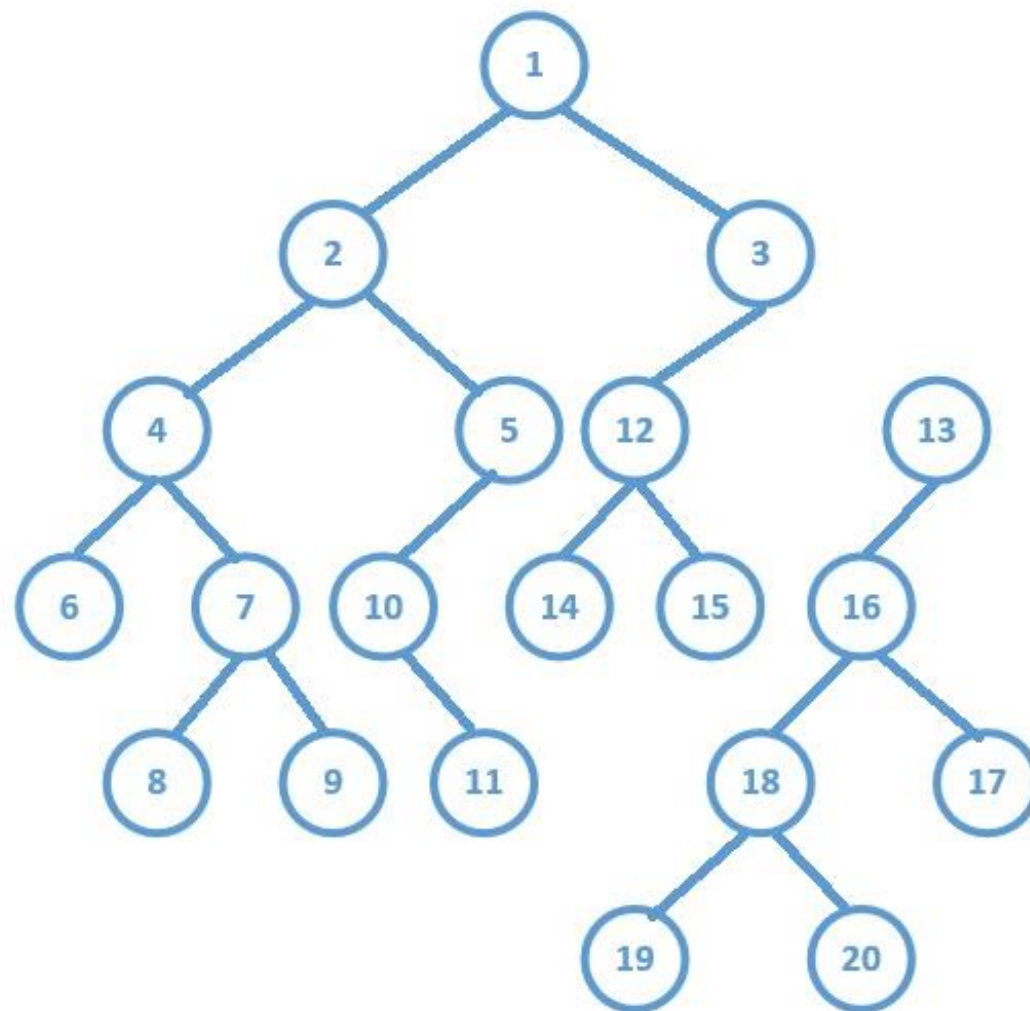


- 上节课讲的三种控制结构，
可以互相包含
- 这节课讲的数据结构，
绝不可以互相包含



作业

- 【习题1】在只有一个单元辅助存储空间的情况下将一个长度为10的数组中所有元素进行倒序排列，看需要多少次操作。
- 【习题2】分别用先序、中序、后序遍历右图中二叉树，写出结果。



谢谢

