

welcome

树与二叉树

2018/10/23

Tivility/魏牧远

heu2016201621

tivility@outlook.com

- 内容

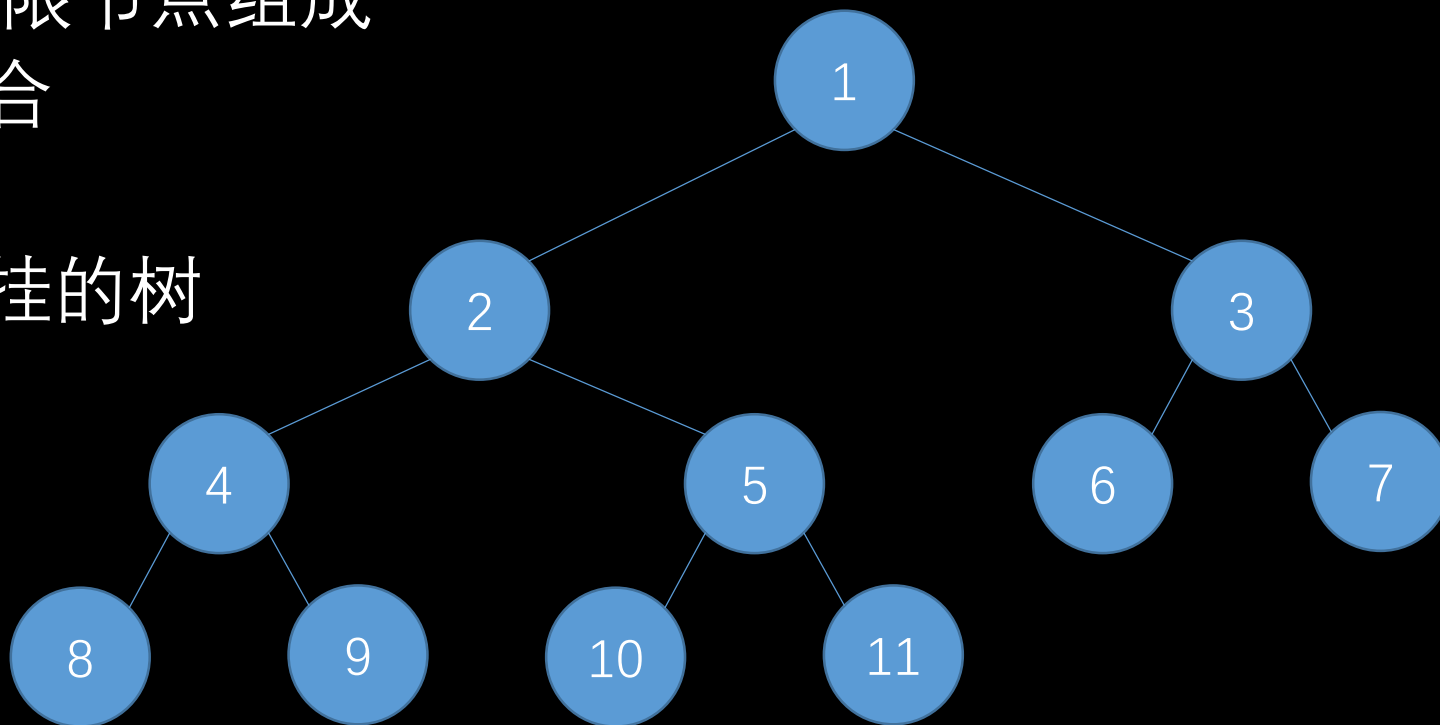
- 定义
- 相关术语
- 二叉树
- 树的遍历
- Huffman编码

- 从前有颗高高的树…
- 上面挂了很多…

- 树(Tree)

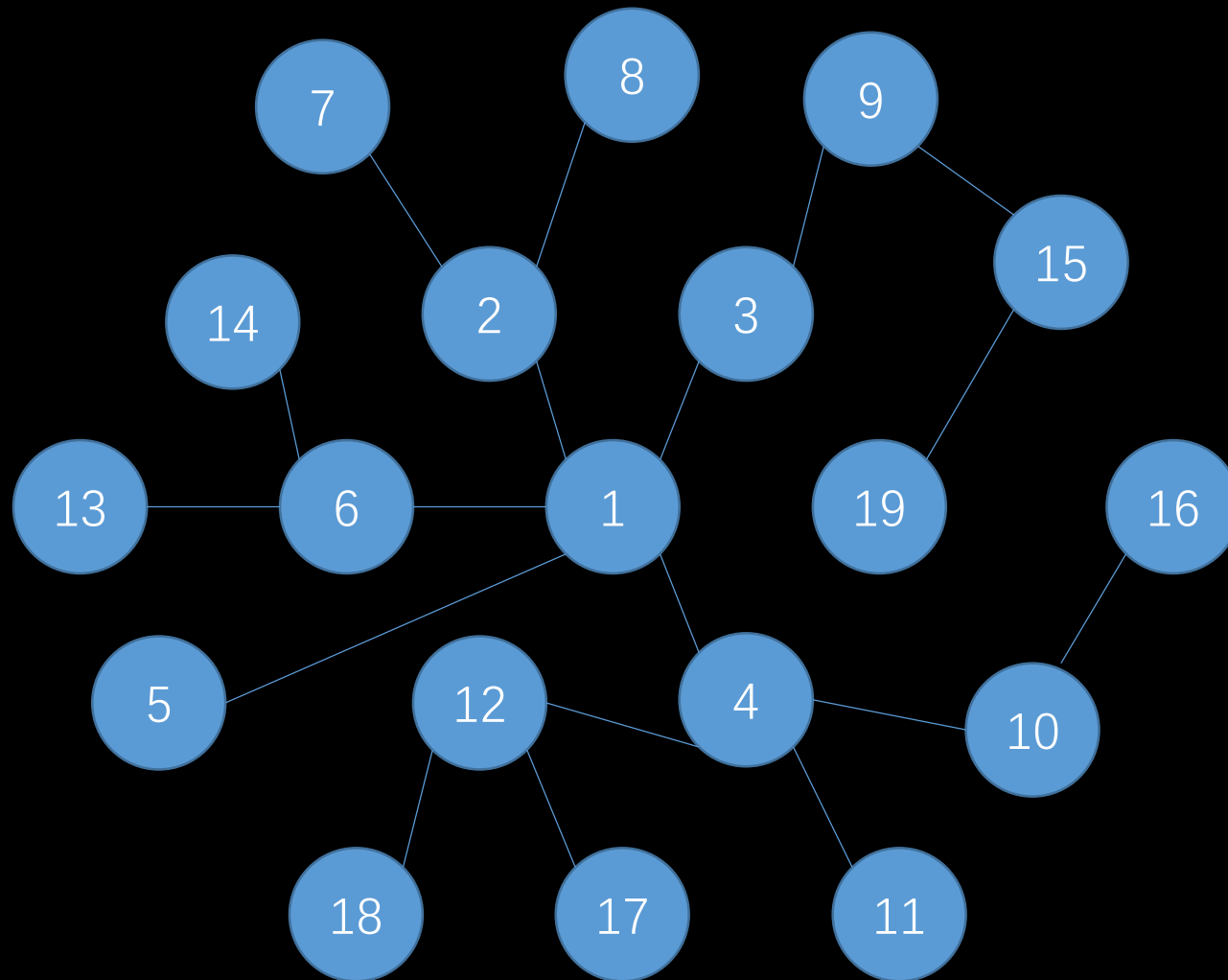
- 由 n ($n > 0$) 个有限节点组成
- 有层次关系的集合

- 看起来像一棵倒挂的树
- 根朝上, 叶朝下



- 树(Tree)

- 这也是一棵树



• 术语

- 节点的度
- 树的度
- 叶节点或终端节点
- 非终端节点或分支节点
- 父亲节点或父节点
- 孩子节点或子节点
- 兄弟节点
- 节点的层次
- 深度
- 高度
- 堂兄弟节点
- 节点的祖先
- 子孙
- 森林

• 术语

- 节点的度：一个节点含有的子树的个数称为该节点的度；
- 树的度：一棵树中，最大的节点的度称为树的度；
- 叶节点或终端节点：度为零的节点；
- 非终端节点或分支节点：度不为零的节点；
- 父亲节点或父节点：若一个节点含有子节点，则这个节点称为其子节点的父节点；
- 孩子节点或子节点：一个节点含有的子树的根节点称为该节点的子节点；
- 兄弟节点：具有相同父节点的节点互称为兄弟节点；
- 节点的层次：从根开始定义起，根为第1层，根的子节点为第2层，以此类推；
- 深度：对于任意节点n,n的深度为从根到n的唯一路径长，根的深度为0；
- 高度：对于任意节点n,n的高度为从n到一片树叶的最长路径长，所有树叶的高度为0；
- 堂兄弟节点：父节点在同一层的节点互为堂兄弟；
- 节点的祖先：从根到该节点所经分支上的所有节点；
- 子孙：以某节点为根的子树中任一节点都称为该节点的子孙。
- 森林：由 m ($m \geq 0$) 棵互不相交的树的集合称为森林

- 二叉树(Binary Tree)

- 每个节点最多只有两个分支
- 不存在分支度大于2的节点
- 树结构
- 左子树 & 右子树
- 有左右次序
- 不能随意颠倒

- 二叉树(Binary Tree)

- 满二叉树
- 完全二叉树
- 二叉查找树
 - 二叉排序树
 - 二叉搜索树
- 平衡二叉树

- 满二叉树(Full Binary Tree)
 - 二叉树
 - 深度为k
 - 有 $2^{k+1} - 1$ 个节点
 - 每一层上的节点数都是最大节点数

- 完全二叉树

(Complete Binary Tree)

- 二叉树
- 除最后一层外，若其余层都是满的
- 最后一层
 - 或者是满的
 - 或者是在右边缺少连续若干节点

二叉查找树 (Binary Sort Tree)

- 若左子树不空
 - 左子树上所有结点的值均小于它的根结点的值
- 若右子树不空
 - 右子树上所有结点的值均大于或等于根结点的值
- 左、右子树也分别为二叉排序树

- 平衡二叉树

(Balanced Binary Tree)

- 结构平衡的二叉搜索树:
 - 叶节点高度差的绝对值不超过1
 - 左右两个子树都是一棵平衡二叉树
- AVL
- R-B tree
- Treap

- 存储
 - 顺序存储
 - 完全二叉树
 - 链式存储
 - 空间更大

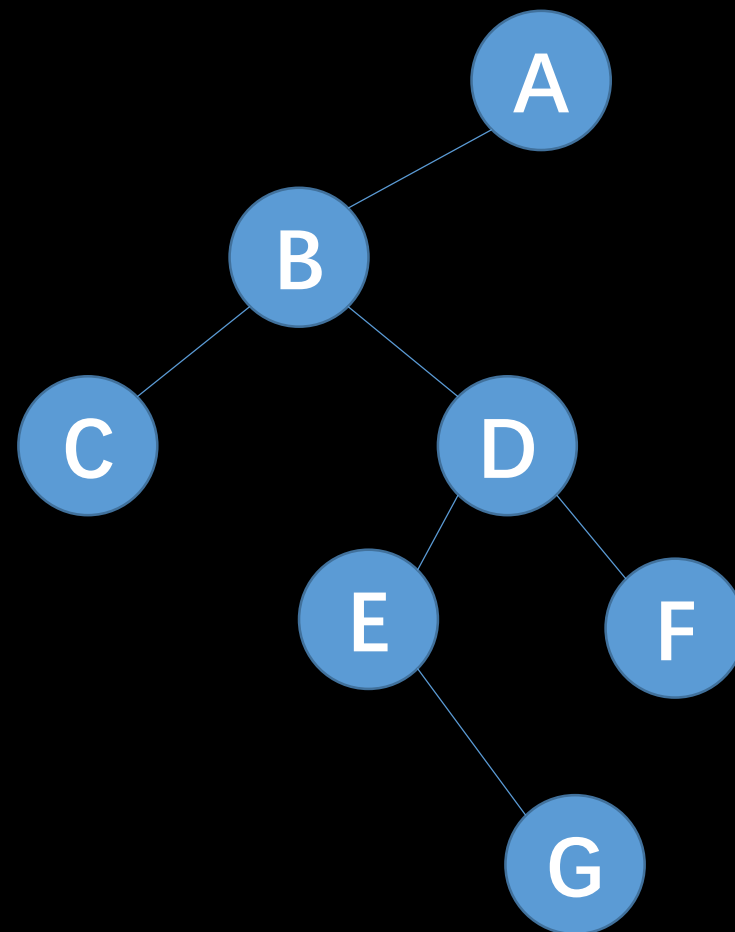
- Do It Yourself!

- 顺序存储

- 链式存储

- 遍历

- 前序遍历：根左右
- 中序遍历：左根右
- 后序遍历：左右根
- 层次遍历：按层遍历



- Huffman 树

1. 将每一个字符作为一个节点加入队列
2. 此时字符节点权重为字频
3. 取出队列中最小权重的两个节点
4. 新建一个节点作为这两个节点的父节点
5. 父节点的权重为子节点权重之和
6. 将此父节点放入原来的队列
7. 重复步骤3 - 6直至队列长度为1
8. 此节点即为Huffman树的根节点

- 习题:
- hdu1053

- END
- 1024程序员节快乐