# 5.二叉树

## 树

Two roads diverged in a yellow wood

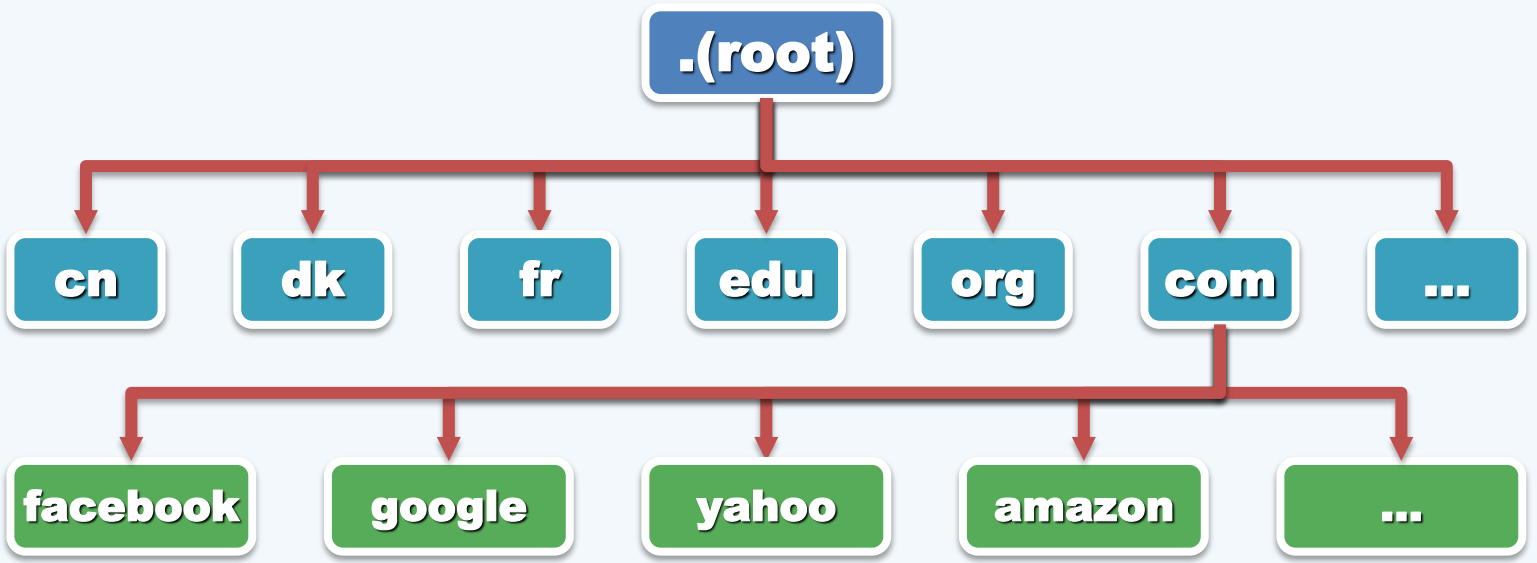
And sorry I could not travel both

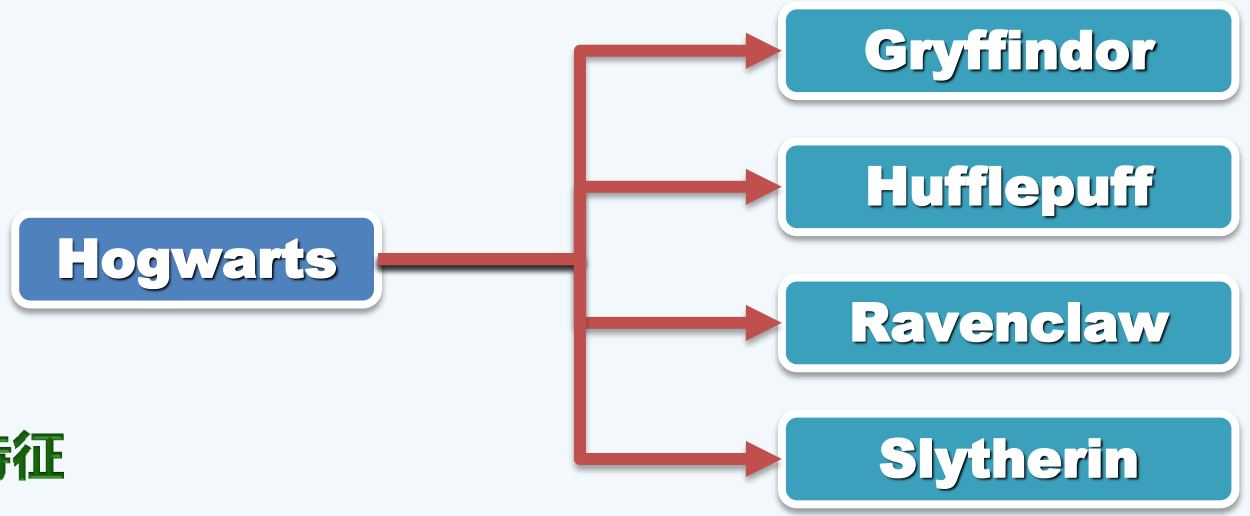
邓俊辉

deng@tsinghua.edu.cn

# 动机

- 【应用】层次结构的表示
  - 表达式
  - 文件系统
  - URL ...
- 【数据结构】综合性
  - 兼具Vector和List的优点
  - 兼顾高效的查找、插入、删除
- 【半线性】
  - 不再是简单的线性结构，但
  - 在确定某种次序之后，具有线性特征

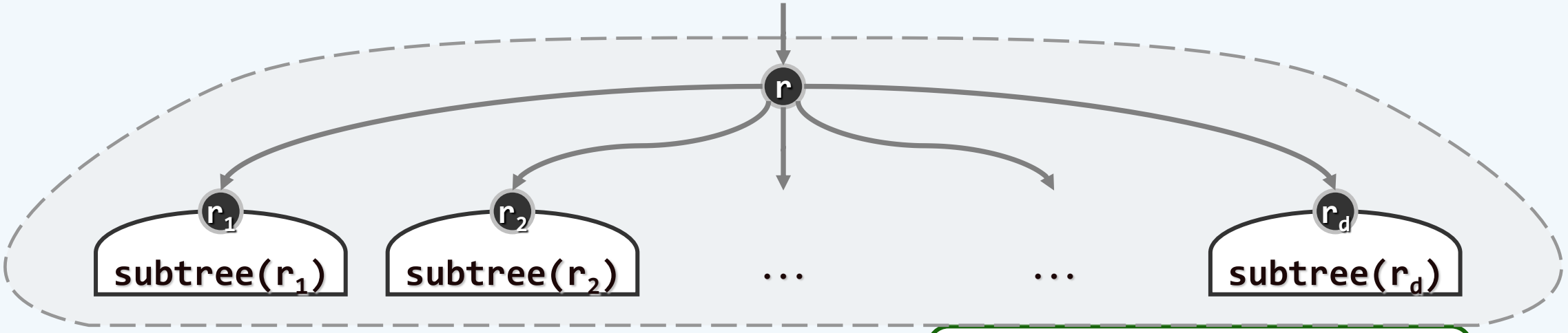


# 有根树

- 树是特殊的图$T = (V, E)$，节点数$|V| = n$，边数$|E| = e$
- 指定任一节点$r \in V$作为**根**后，T即称作**有根树**（rooted tree）
- 若：$T_1, T_2, \ldots T_d$为有根树

  则：$T = ( (\cup V_i) \cup \{r\}, (\cup E_i) \cup \{ <r, r_i> \mid 1 \le i \le d \} )$也是
- 相对于T，$T_i$称作以$r_i$为根的**子树**（subtree rooted at $r_i$），记作$T_i = subtree(r_i)$



# 有序树

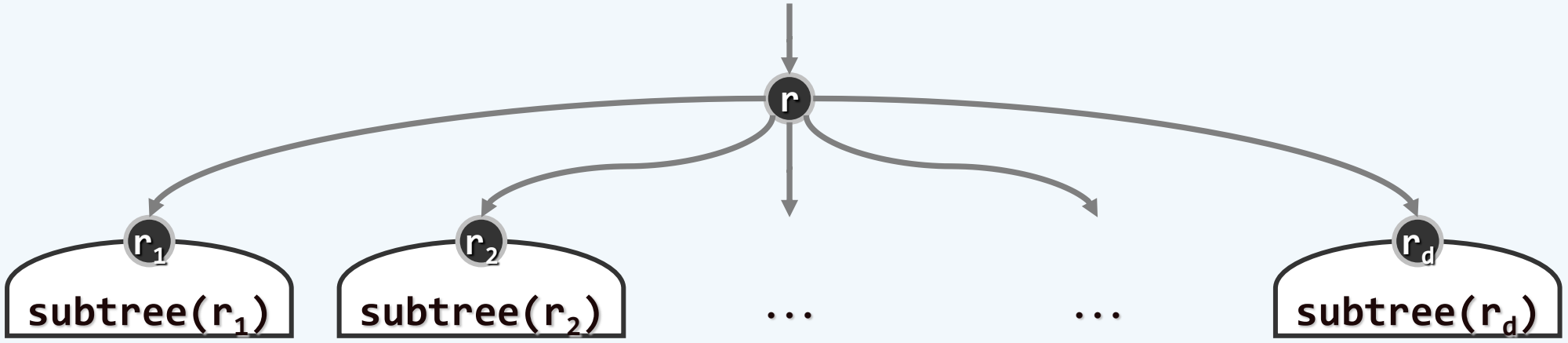
- $r_i$称作r的**孩子**（child），$r_i$之间互称**兄弟**（sibling）

  r为其**父亲**（parent），d = degree(r)为r的（出）**度**（degree）

- **可归纳证明**：e = $\sum_{r \in V} degree(r)$ = n - 1 = **Θ(n)**

  **故在衡量相关复杂度时，可以n作为参照**

- **若指定$T_i$作为T的第i棵子树，$r_i$作为r的第i个孩子，则T称作有序树**（ordered tree）



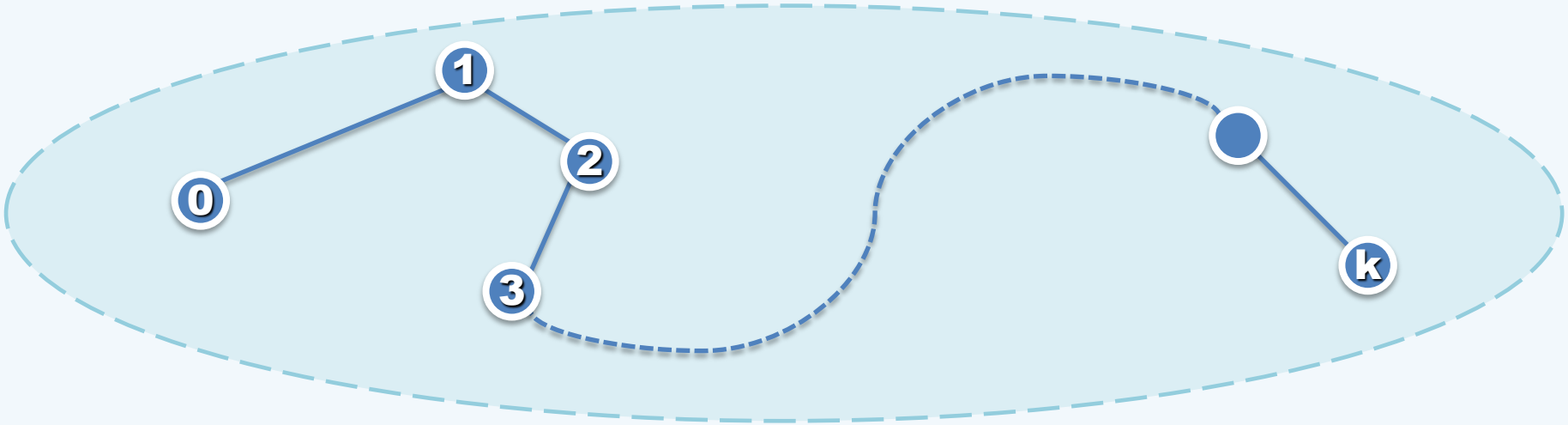
# 路径 + 环路

❖ V中的k+1个节点，通过E中的k条边依次相联，构成一条**路径**（path） //亦称**通路**

$$\pi = \{ (v_0, v_1), (v_1, v_2), \cdots\cdots , (v_{k-1}, v_k) \}$$

❖ 路径**长度**：$|\pi|$ = 边数 = k //早期文献，或以节点数为长度

❖ **环路**（cycle/loop）：$v_k = v_0$



# 连通 + 无环

- 节点之间均有路径，称作**连通图**（connected）

  不含环路，称作**无环图**（acyclic）

- 树：　**无环连通图**

  **极小连通图**

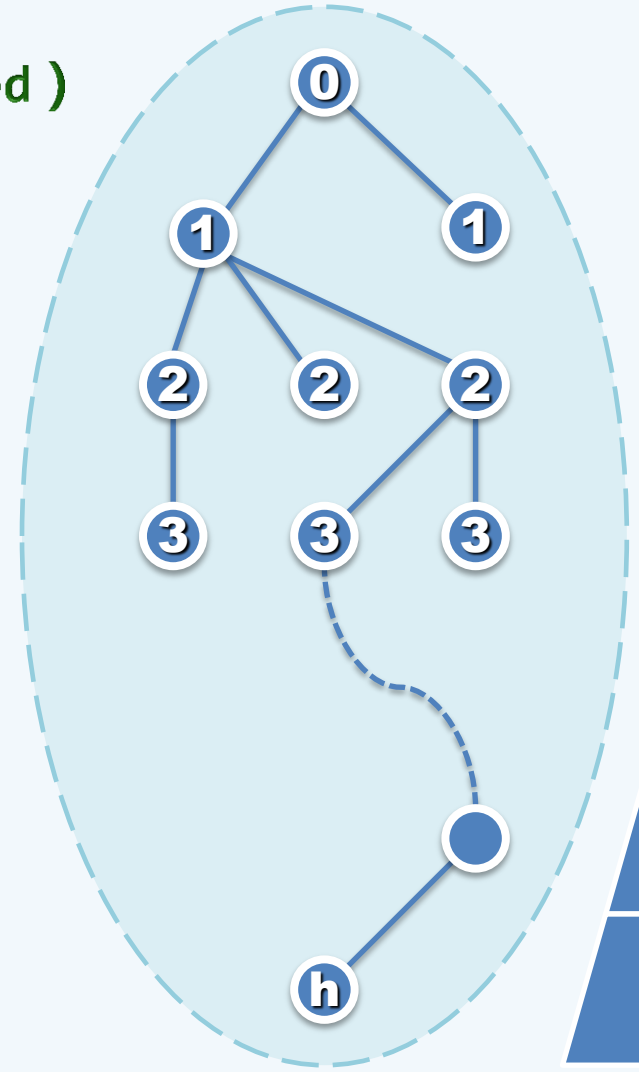
  **极大无环图**

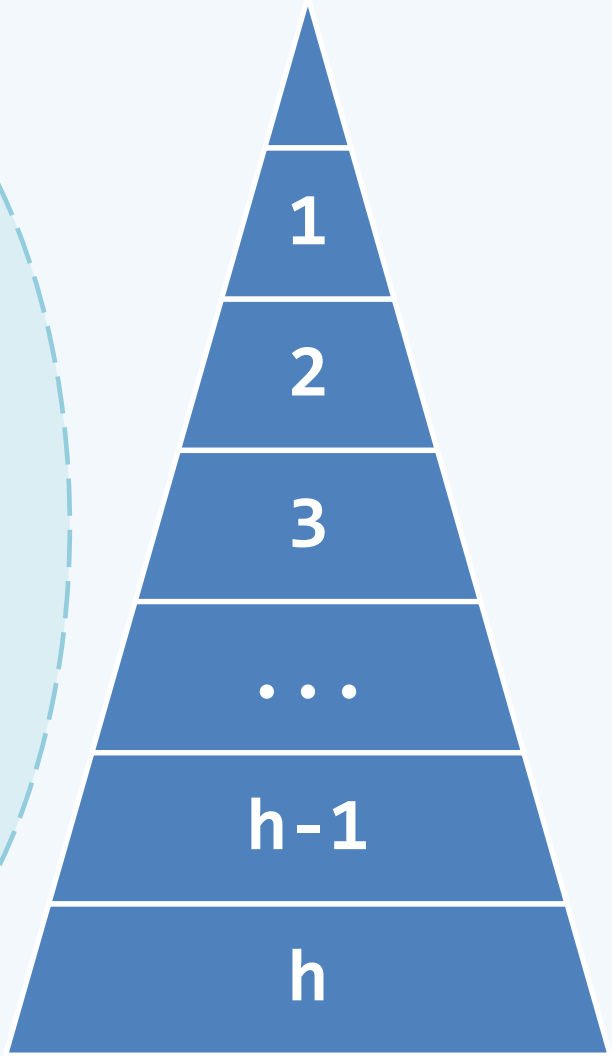
- 故：　任一节点v与根之间存在**唯一**路径

  path(v, r) = path(v)

- 于是：以|path(v)|为指标

  可对所有节点做**等价类**划分...



# 深度 + 层次

- 不致歧义时，路径、节点和子树可**相互指代**

  path(v) ~ v ~ subtree(v)

- v的**深度**：depth(v) = |path(v)|

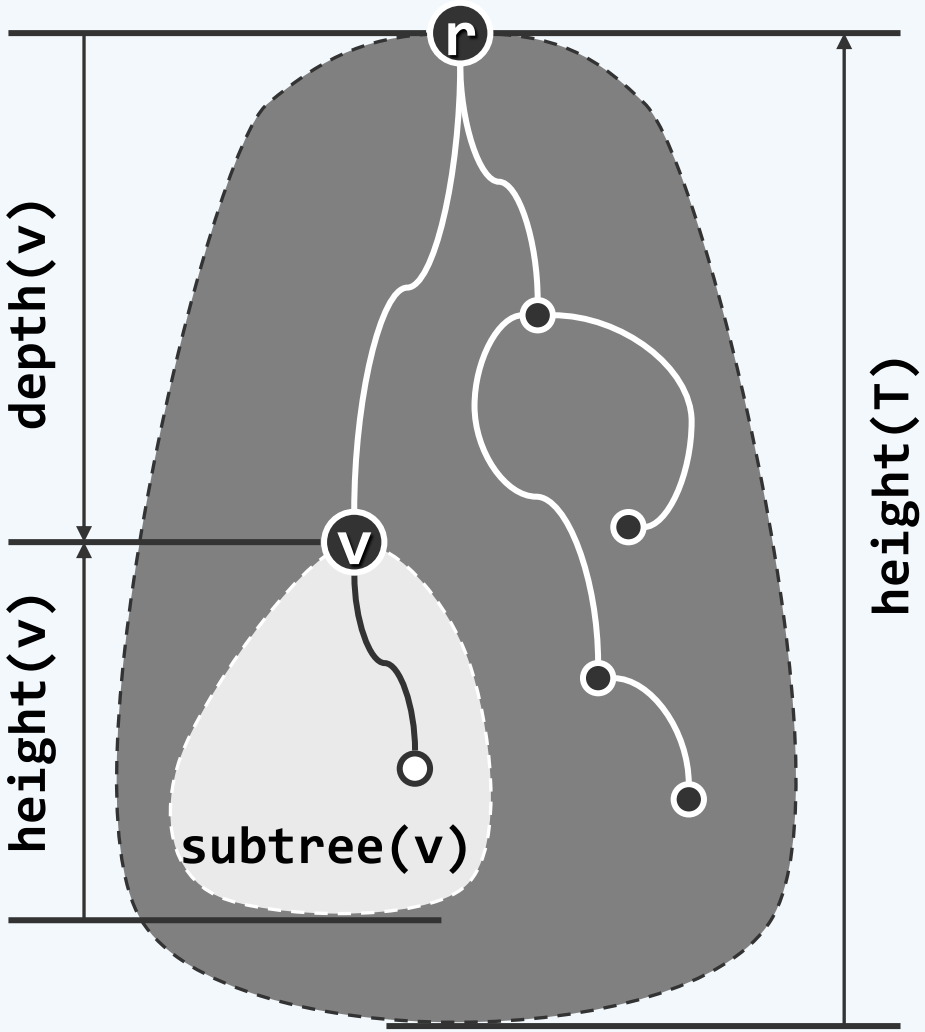
- path(v)上节点，均为v的**祖先**（ancestor）

  v是它们的**后代**（descendent）

- 其中，除自身以外，是**真**（proper）祖先/后代

- **半线性**：在任一深度

  v的祖先/后代若存在，则**必然**/**未必**唯一



# 深度 + 层次

- 根节点是所有节点的**公共祖先**，深度为0
- 没有后代的节点称作**叶子**（leaf）
- 所有叶子深度中的最大者

  称作（子）树（根）的**高度**

  height(v) = height( subtree(v) )
- 特别地，空树的高度取作**-1**
- depth(v) + height(v) ≤ height(T)

  何时取等号？