

5. 二叉树

(a) 树

Two roads diverged in a yellow wood

And sorry I could not travel both

- Robert Frost, 1915

邓俊辉

deng@tsinghua.edu.cn

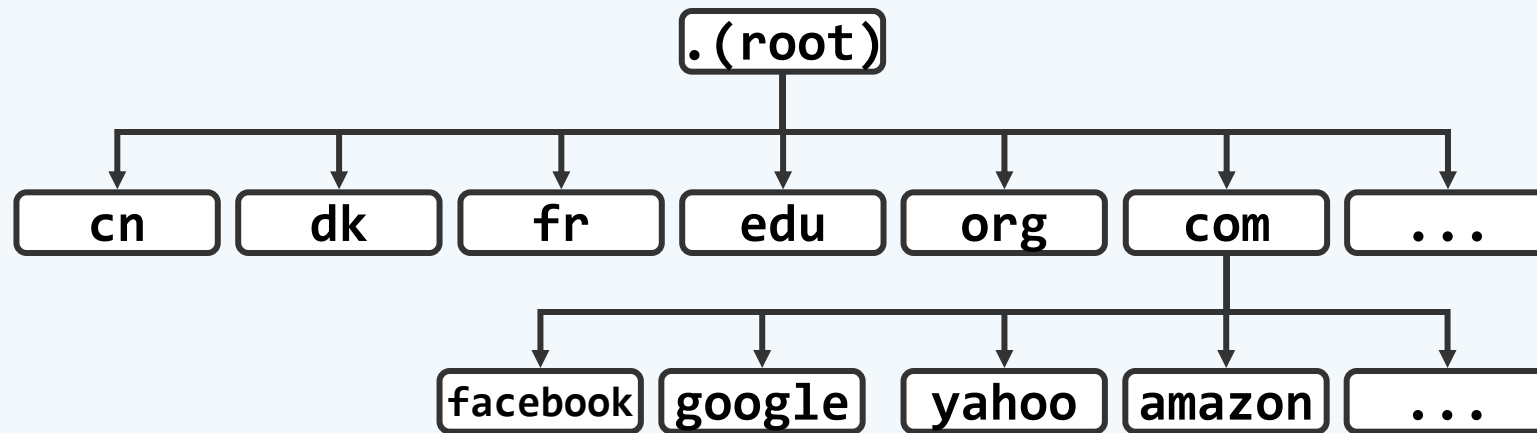
应用

❖ 层次关系的表示

表达式

文件系统

URL ...



❖ 综合性：

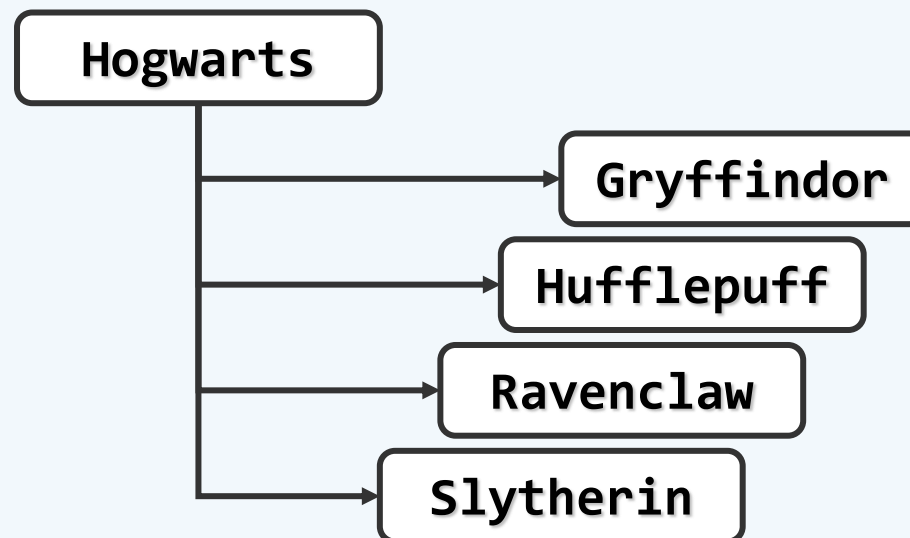
兼具Vector和List的优点

兼顾高效的查找、插入、删除

❖ 半线性：

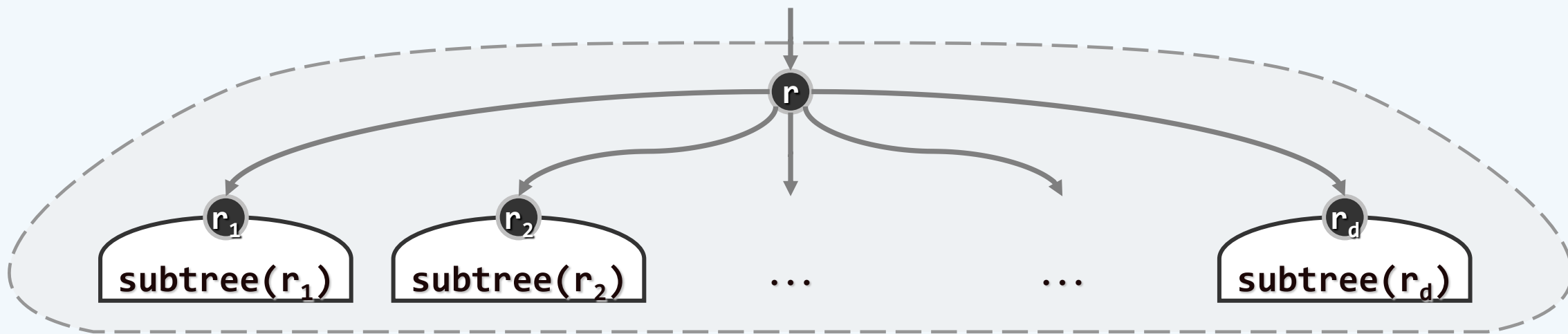
不再是简单的线性结构，但

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, \dots, T_d 为有根树
则: $T = ((\cup V_i) \cup \{r\}, (\cup E_i) \cup \{ \langle r, r_i \rangle \mid 1 \leq i \leq d \})$ 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 $T_i = \text{subtree}(r_i)$



有序树

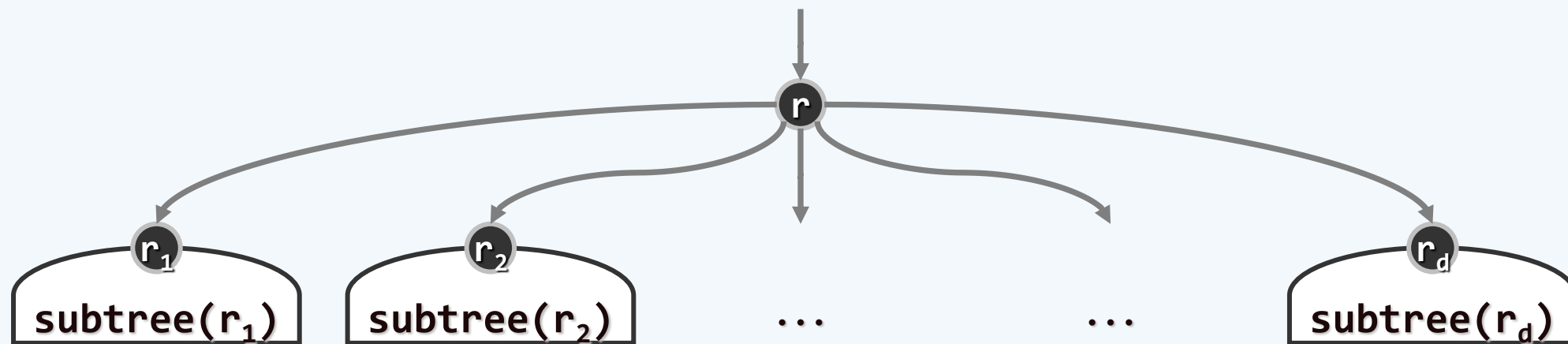
❖ r_i 称作 r 的 **孩子** (child), r_i 之间互称 **兄弟** (sibling)

r 为其 **父亲** (parent), $d = \text{degree}(r)$ 为 r 的 (出) **度** (degree)

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

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)

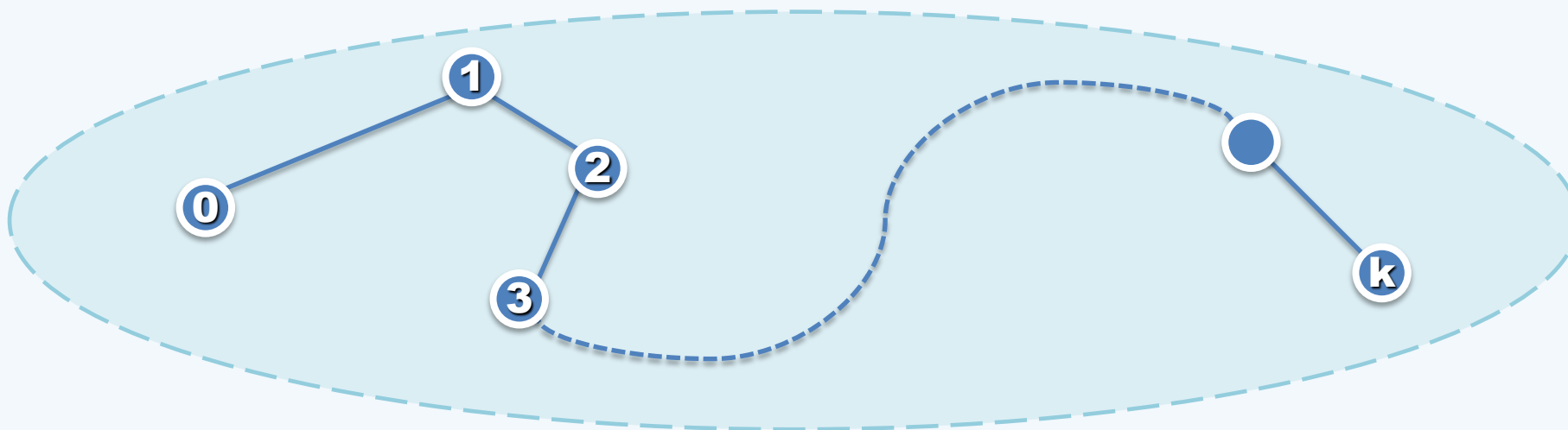
// 亦称 **通路**

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

❖ **路径长度** : $|\pi| = \text{边数} = k$

// 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

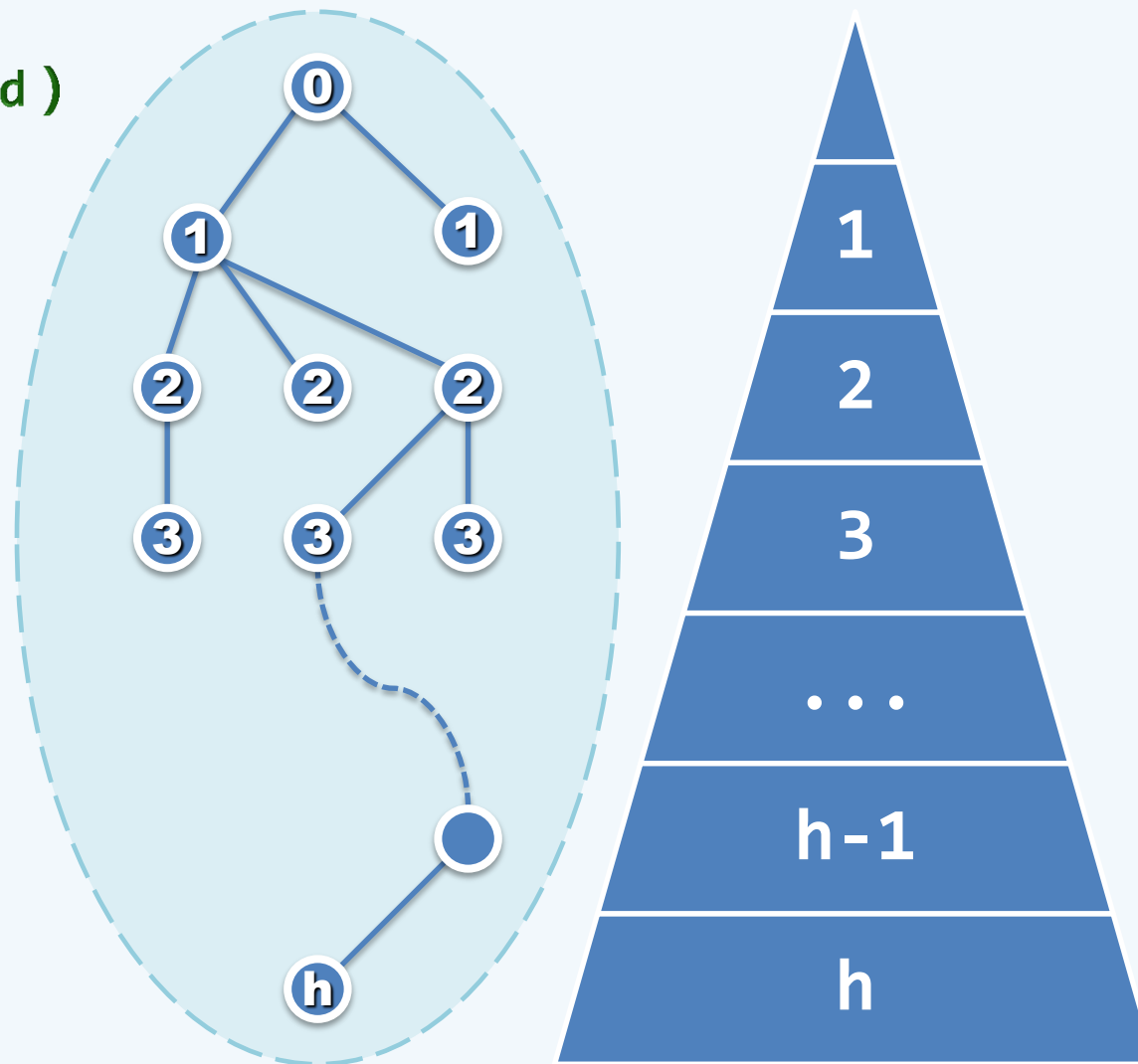
极大无环图

❖ 故：任一节点 v 与根之间存在**唯一**路径

$$\text{path}(v, r) = \text{path}(v)$$

❖ 于是：以 $|\text{path}(v)|$ 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

$$\text{path}(v) \sim v \sim \text{subtree}(v)$$

❖ v 的深度： $\text{depth}(v) = |\text{path}(v)|$

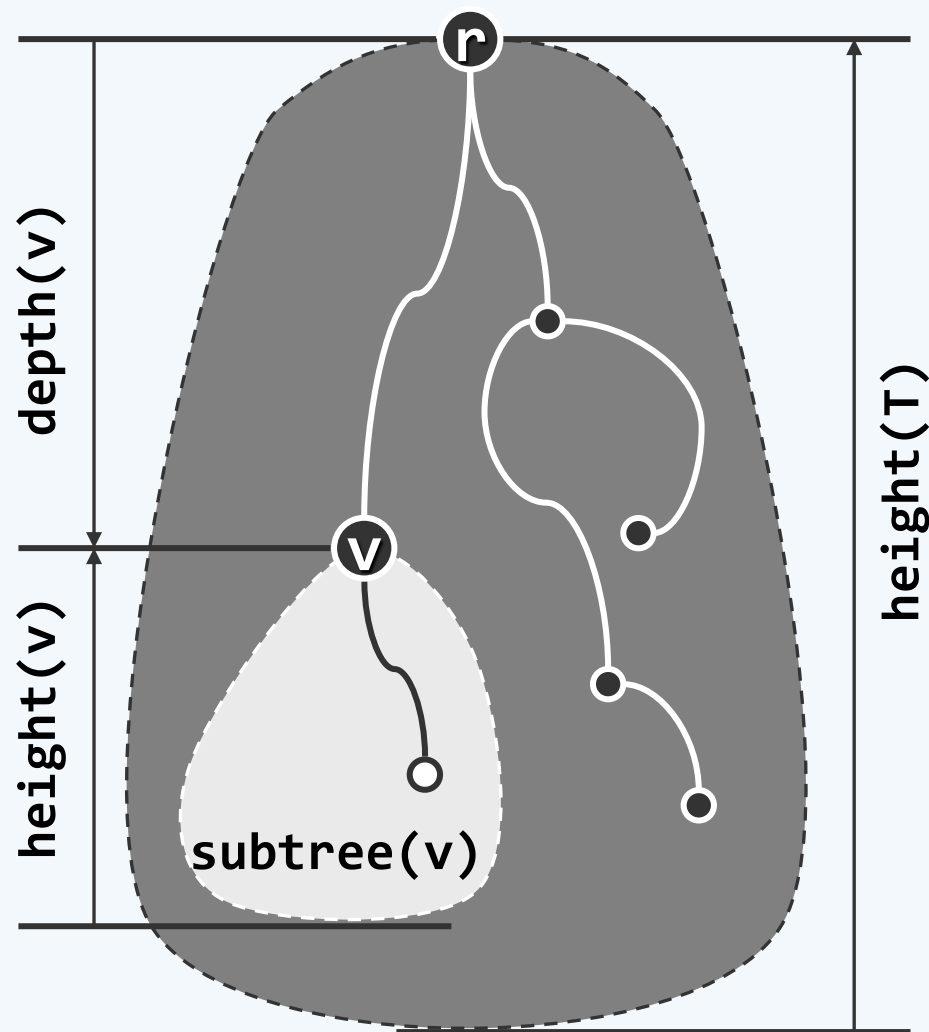
❖ $\text{path}(v)$ 上节点，均为 v 的祖先（ancestor）

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

$$\text{height}(v) = \text{height}(\text{subtree}(v))$$

❖ 特别地，空树的高度取作-1

❖ $\text{depth}(v) + \text{height}(v) \leq \text{height}(T)$

何时取等号？

