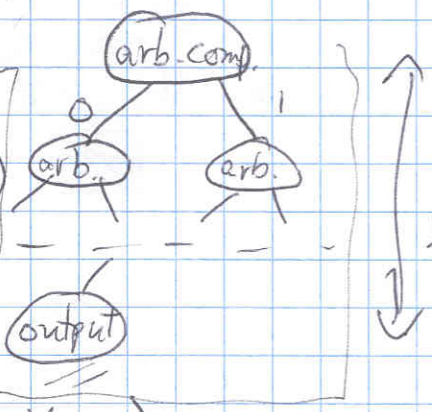
6.897
L3.3
Feb. 19, 2003

Decision-tree model (e.g. comparison model)

- algorithm is a (large) tree:
- each node does arb. computation (over input, data structure, etc.) but outputs binary decision
- leaf specifies the output.
- (which can only depend on unique root-to-leaf path there)



len(root-leaf path) = running time

Lower bound: Any dynamic DS supporting Insert & Successor requires $\Omega(\lg n)$ amortized time in worst case in decision-tree model

Proof: Reduction from Sort$(a_1, a_2, \ldots, a_n)$:

$$\left.\begin{array}{l} \text{Insert}(a_1) \\ \text{Insert}(a_2) \\ \vdots \\ \text{Insert}(a_n) \end{array}\right\} n$$

$$\left.\begin{array}{l} \text{print } x = \text{Successor}(-\infty) \\ \text{print } x = \text{Successor}(x) \\ \vdots \\ \text{print } x = \text{Successor}(x) \end{array}\right\} n$$

$\therefore$ Sort $\leq$ Insert & Successor

Sort requires $\Omega(n \lg n)$ because need to determine a permutation, there are $n!$ of them, and only learn 1 bit per time step. (Kolmogorov). //