CIT 596
Lower Bounds in the Comparison Tree Model

Penn Engineering

COMPARISON TREE MODEL

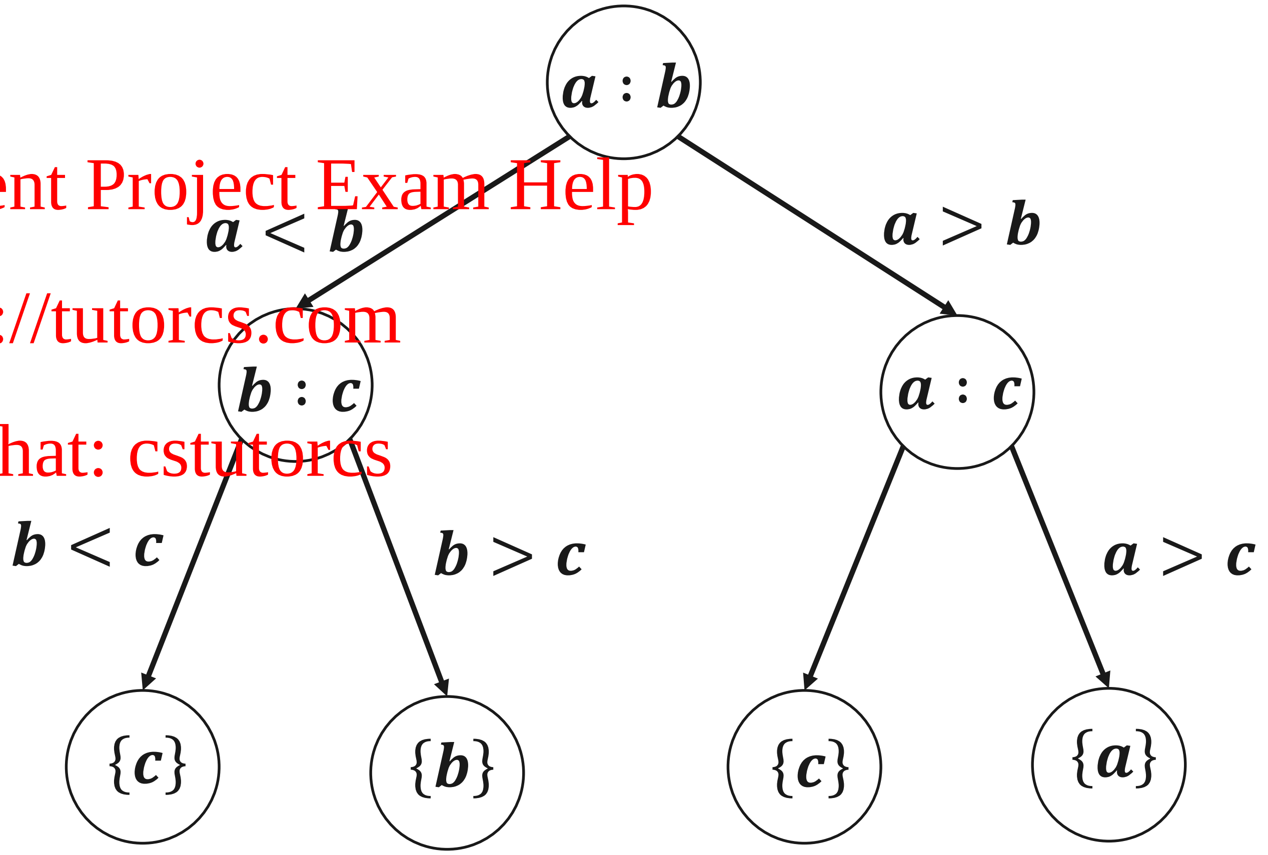
- For many sorting and selection algorithms, the majority of the work is comparing input elements.
- For such algorithms, it is reasonable to ignore all other operations and count just the comparisons.
- Can also represent the behavior of the algorithm as a tree:
  - Root labeled by the first comparison of two elements, say x and y
  - Left subtree represents behavior if $x < y$
  - Right subtree represents behavior if $x > y$
- Each internal node is labeled with a comparison:
  - The two children correspond to the two results
- Leaf nodes labeled by final answers
- Height of the tree corresponds to worst-case number of comparisons

EXAMPLE OF COMPARISON TREE FOR "MAX OF 3"

Input: three distinct numbers $\boldsymbol{a, b, c}$

Output: the maximum of $\boldsymbol{a, b, c}$

- Leaf nodes labeled with answers. Thus each leaf node is labeled with a single element.



COMPARISON TREE FOR SORTING

- Example: comparison tree for sorting n numbers

- Suppose the input is $x_1, x_2, \ldots, x_n$

- What are the possible answers?

  - Each answer corresponds to a certain ordering of the input.
  - There are $n!$ possible orderings/answers.

  - There must be at least one leaf for each answer.

SORTING LOWER BOUND

- For a binary tree to have $\boldsymbol{n}!$ leaves, what is the required height $\boldsymbol{h}$?
  - It must be big enough that $2^h \geq n!$
  - Taking logs:

$$h \geq \log n! = \Omega(n \log n)$$

- Sorting in the comparison tree model requires $\Omega(n \log n)$ comparisons.
- Information-theoretic lower bound, obtained by counting answers.

MERGING TWO SORTED LISTS

- Given $a_1 < a_2 < \cdots < a_n$ and $b_1 < b_2 < \cdots < b_n$
- Want to produce a merged list of length $2n$
- Comparison tree algorithm. How many answers?
  - Each answer corresponds to choosing the n positions that elements from the a-list will occupy.