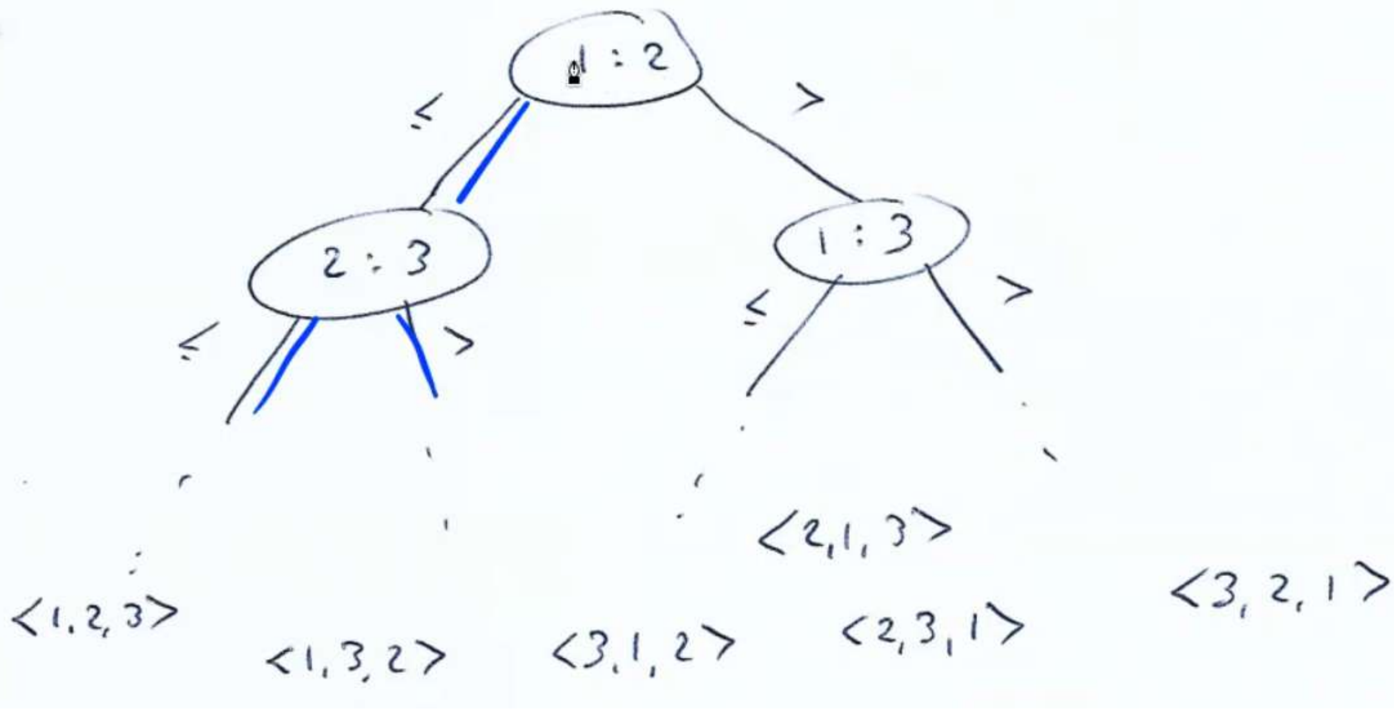
Chapter 8 Sorting in Linear Time

8.1 Lower bounds for sorting.

Consider the following decision tree
for Insertion Sort :


1 : 2
≤
>
2 : 3
1 : 3
<1, 2, 3>
<1, 3, 2>
<3, 1, 2>
<2, 1, 3>
<2, 3, 1>
<3, 2, 1>

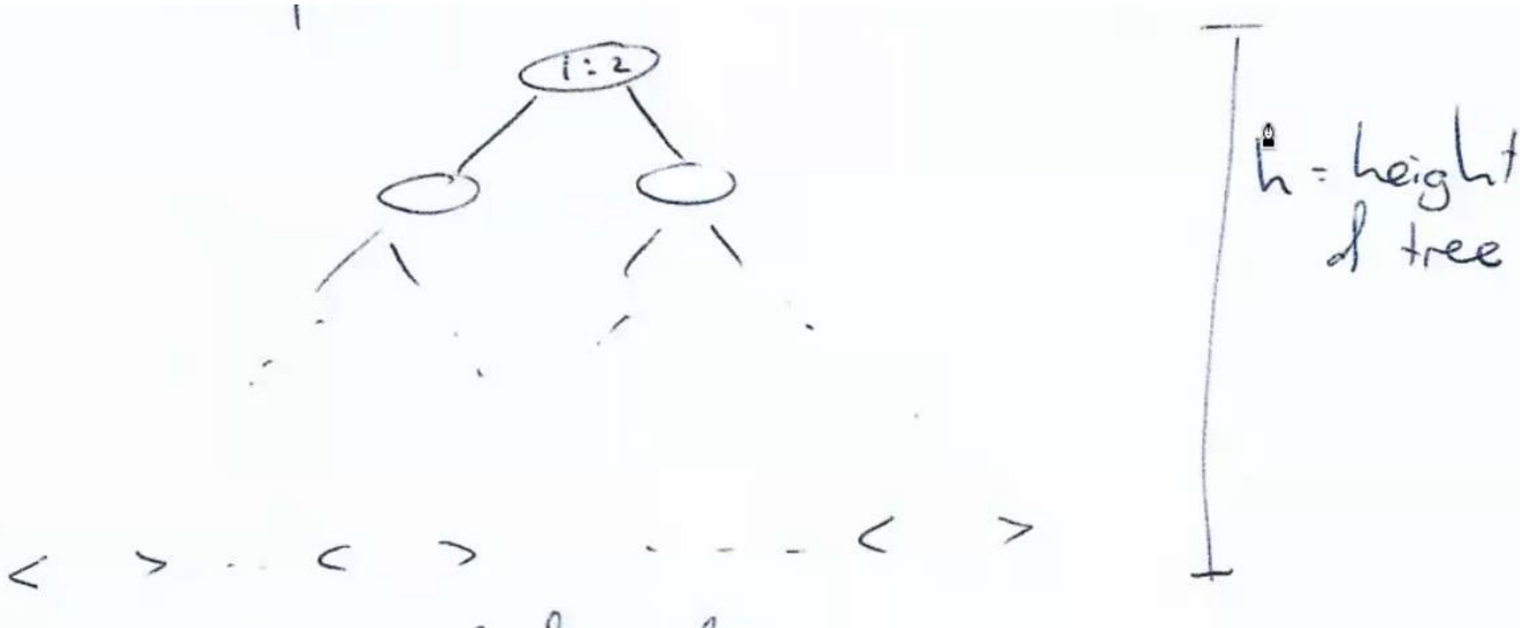
- The leaf nodes are the permutations of the input.
- Every possible permutation must be represented at at least one leaf node.
- Any path from root to a leaf represents an execution of the sort algorithm.

- The height of tree is the longest path from root to a leaf node.

Theorem 8.1

Any comparison sort algorithm requires $\Omega(n \log n)$ comparisons in the worst case.

Proof: Draw the decision tree for the algorithm on an input of size n.



l = # leaf nodes

ℓ = # leaf nodes

We know: $\ell \leq 2^h$ and $\ell \geq n!$

Thus $n! \leq \ell \leq 2^h$

$\therefore \quad n! \leq 2^h$

$\therefore \quad \log(n!) \leq h$

Recall $\log(n!) = \Omega(n \log n)$

Thus $h = \Omega(n \log n)$

so the worst case number of comparisons is $\Omega(n \log n)$