Let $A = [a_1, a_2, \cdots, a_n]$ be a list of n distinct elements in sorted (ascending) order. Given any of the $n!$ permutations of this list, a sorting algorithm must be able to output precisely the one order in which the elements are sorted.

1. We build a decision tree with nodes corresponding to the comparisons made by the algorithm. Does each input have a different decision tree associated with it? How about the various comparison based sorting algorithms?

2. Let v be a leaf in the decision tree of a sorting algorithm. The nodes along a root-to-leaf path represent the comparisons performed to sort an input, with sufficient information to perform the sorting obtained by the time we reach the leaf node.

   If the inputs corresponding to two permutations $\sigma(A)$ and $\sigma'(A)$ get sorted by following the paths from the root to v, then show that $\sigma(A) = \sigma'(A)$.

   Conclude that the decision tree has $n!$ leaf nodes.

3. Show that a binary tree with $n!$ nodes has height at least $\Omega(\log(n!))$.

4. Show that $\log(n!) = \Theta(n \log n)$. Note that to show a tight bound, you need to show both the upper and the lower bounds.