Lower bound for Worst-Case Behavior of any Sorting Algorithm that uses Comparison-Swap model

Lemma 1: To every deterministic algorithm for sorting n distinct keys there corresponds a valid, binary decision tree containing exactly $n!$ leaves.

Proof: There is a valid decision tree corresponding to any algorithm for sorting n keys. When all the keys are distinct, the result of a comparison is always "<" or ">". Therefore, each node in that tree has at most two children, which means that it is a binary tree. Next we show that it has $n!$ leaves. Because there are $n!$ different inputs that contain n distinct keys and because a decision tree is valid for sorting n distinct keys only if it has a leaf for every input, the tree has at least $n!$ leaves. Because there is a unique path in the tree for each of the $n!$ different inputs from the root of the tree and because every leaf in a decision tree must be reachable, the tree can have no more than $n!$ leaves. Therefore, the tree has exactly $n!$ leaves.

Lemma 2: The worst-case number of comparisons done by a decision tree is equal to its depth.

Proof: Given some input, the number of comparisons done by a decision tree is the number of internal nodes on the path followed for that input. The number of internal nodes is the same as the length of the path. Therefore, the worst-case number of comparisons done by a decision tree is the length of the longest path to a leaf, which is the depth of the decision tree.

Now we need only to find a lower bound on the depth of a binary tree containing $n!$ leaves to obtain a lower bound for the worst-case behavior. The required lower bound is found by means of the following lemma.

Lemma 3: If m is the number of leaves in a binary tree and d is the depth, then

$$d \geq \lceil \log m \rceil$$

Proof: Use induction on d.

Theorem: Any deterministic algorithm that sorts n distinct keys using the comparison-swap model must in the worst case do at least

$$\lceil log\,(n!) \rceil \text{ comparisons of keys}$$

<u>Proof:</u> By Lemma 1, to any such algorithm there corresponds a valid, binary decision tree containing n! leaves. By Lemma 3, the depth of that tree is greater than or equal to $\lceil log\, n! \rceil$.

The theorem now follows, because Lemma 2 says that any decision tree's worst-case number of comparisons is given by its depth.

Lemma 4: For any positive integer n,

$$log(n!) \geq n\, log\, n - 1.45n$$