Comparison Sort Lower Bound

Daniel Axline

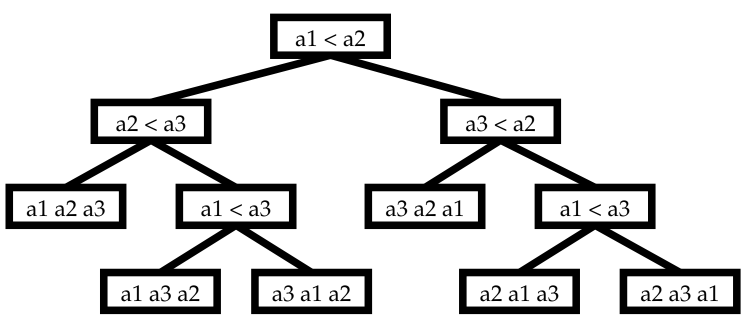
Edited by Chuck Cusack

These notes are based on section 9.1 of [1] and lectures from CSCE423/823, Spring 2001.

A sorted order derived by any comparison sort algorithm is obtained by sequential comparisons of elements. Since comparisons have only two outcomes, comparison sort algorithms imply an associated binary decision tree. We can use what we know about binary trees to determine that the worst case for any comparison sort is at least $\Omega(n \log n)$[1].

1 Decision Trees

Every comparison sort has an associated decision tree. The nodes of the decision tree correspond to comparisons made by the algorithm (so different algorithms generally have different decision trees). The root node is the first comparison made by the algorithm. If a comparison is false, the right child node is the next comparison; if a comparison is true, the left child node is the next comparison. Thus the leaves of a decision tree are all the possible outcomes of for the corresponding comparison sort algorithm. Since there are $n!$ possible orderings of an array of size n, any decision tree corresponding to a correct sorting algorithm must have at least $n!$ leaves[2], since it must be able to sort given any of the orderings. Here is an example decison tree for sorting three elements:



[1] All logs are base 2.

[2] Since there may be multiple ways to determine the correct order, there may be multiple leaves corresponding to the same order, thus there may be more than $n!$ leaves.

2 Worst Case Lower Bound

Since every comparison sort algorithm has a decision tree, we can compute the the lower bound on any comparison sort by bounding the lowest height of a decision tree. Recall that a binary tree with height h has at most 2^h leaves.

Theorem 1 *A decision tree corresponding to a comparison sort has height* $\Omega(n \log n)$.

Proof: A decision tree corresponding to a comparison sort has at least n! leaves. Therefore $n! \leq 2^h$ where h is the height of the decision tree. Taking logs, we get

$$\log(n!) \leq \log(2^h) = h \log 2 = h$$

From Sterlings approximation we know that $n! \geq (\frac{n}{e})^n$, so

$$h \geq \log(n!) \geq \log(\frac{n}{e})^n = n \log(\frac{n}{e}) = n(\log n - \log e) = \Omega(n \log n)$$

Therefore any decision tree corresponding to a comparison sort has height $\Omega(n \log n)$. $\square$

References

[1] Cormen, Leiserson, and Rivest, *Introduction to Algorithms*, McGraw Hill, 1990.