CSCI 3100

Comparison Sorts

Review & Overview

Last time on Algorithms:

- Counted all operations of insertion sort
- Introduced “asymptotic analysis”:
  - g(n) = O(f(n)) => f(n) is an asymptotic upper bound of g(n)
  - g(n) = Ω(f(n)) => f(n) is an asymptotic lower bound of g(n)
  - g(n) = Θ(f(n)) => f(n) is asymptotically equivalent to g(n)

This time:

- Lower bound on comparison sorts
- Formal proof
- First homework

How Fast Can We Sort?

Insertion sort: $O(n^2)$

Bubble Sort, Selection Sort: $\Theta(n^2)$

Merge sort: $\Theta(n\lg n)$

Quicksort: $\Theta(n\lg n)$ - average

What is common to all these algorithms?

- They all sort by making **comparisons** between the input elements

Comparison Sorts

Compare elements to gain information about an input sequence $\langle a_1, a_2, \ldots, a_n \rangle$

Perform tests: $a_i < a_j$, $a_i \le a_j$, $a_i = a_j$, $a_i \ge a_j$, or $a_i > a_j$ to determine the relative order of a_i and a_j

For simplicity, assume that all the elements are distinct

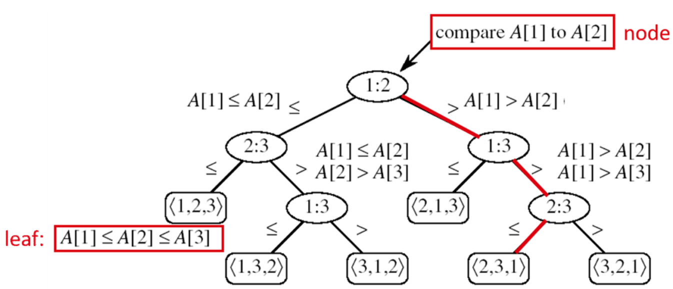
Lower bound on comparison based sorting algorithms

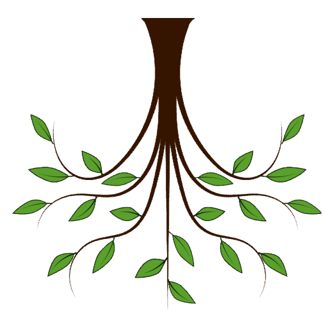
Theorem:
Any comparison sort algorithm requires Ω(n lg(n)) comparisons in the worst case

- True or False?
  - Any comparison sort algorithm will take at least *n lg (n)* comparisons to complete for ALL inputs.
  - For any comparison sort algorithm, there exists an input that will take at least *n lg(n)* comparisons to complete.
  - There is no comparison sort algorithm that will take less than *n lg(n)* comparisons for ALL inputs.

Decision Tree Model

- Represents the comparisons made by a sorting algorithm on an input of a given size.
  - Models all possible execution traces
  - Control, data movement, other operations are ignored
  - Count only the comparisons

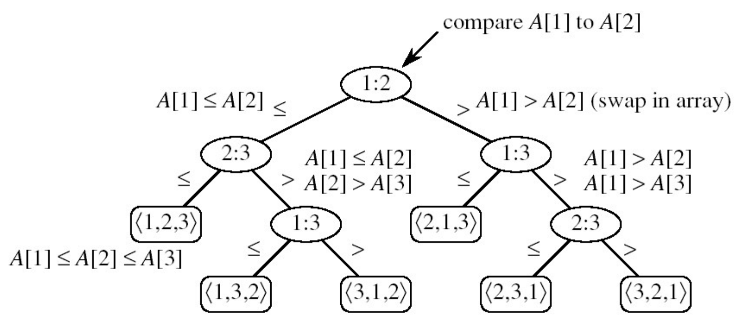




Worst-case number of comparisons?

- Worst-case number of comparisons depends on:
  - the length of the longest path from the root to a leaf

    (i.e., the height of the decision tree)



Lemma

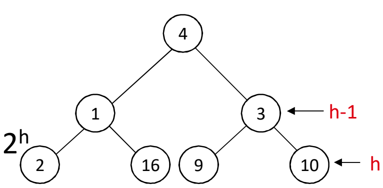
- Binary tree of height h has at most 2^h leaves
- Proof by induction on h

Base Case

- h=0 => tree has one node, which is a leaf
- # leaves = 1 ≤ 2^0 (TRUE)

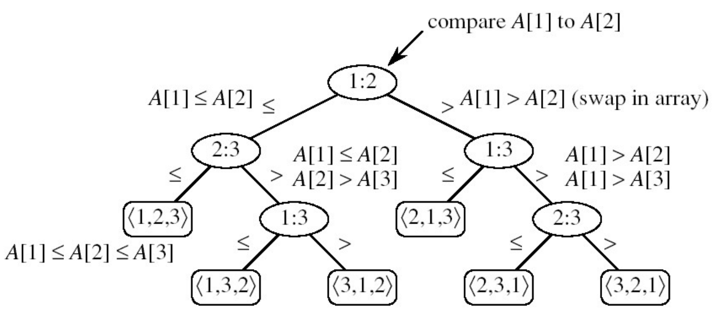
Inductive Step

- Assume true for h-1 (# leaves ≤ 2^{h-1})
- Extend the height of the tree to one more level
- Each leaf becomes a parent to two new leaves
- # leaves at level h = 2*(#leaves at level h-1)≤ $2*2^{h-1} = 2^h$



What is the least number of leaves in a Decision Tree Model?

- All permutations on n elements must appear as one of the leaves in the decision tree: **n!** **permutations**
- At least **n!** leaves



Lower Bound for Comparison Sorts

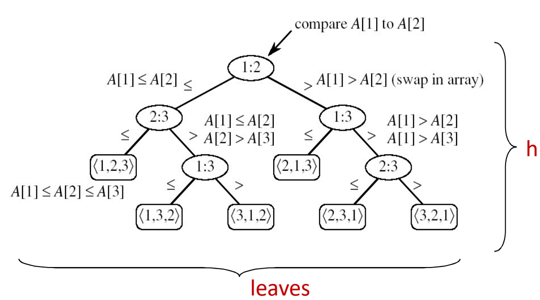
Theorem:
Any comparison sort algorithm requires Ω(n lg(n)) comparisons in the worst case

Proof: How many leaves does the tree have?

- At least n! (each of the n! permutations must appear as a leaf)
- There are at most 2^h leaves (by the previous Lemma)

➔ $n! \leq 2^h$

➔ $h \geq \lg(n!) = \boldsymbol{\Omega(?)}$



lg(n!) = Ω(?)

1. n! ≥ 2^n ➔ lg(n!) ≥ n lg2 = n ➔ lg(n!) = Ω(n) ➔➔➔ n ≤ lg(n!)

- We need a tighter *lower* bound!
- Use Stirling's approximation (3.18):

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)$$

$$\begin{aligned}\lg(n!) &= \lg\sqrt{2\pi n} + \lg\left(\frac{n}{e}\right)^n + \lg\left(1+\Theta\left(\frac{1}{n}\right)\right) \\ &\geq n\lg\left(\frac{n}{e}\right) \geq cn\lg n \qquad \text{for } c = 0.5 \text{ and } n > n_0 = e^2\end{aligned}$$

$$\lg(n!) = \Omega(n\lg n)$$