Sorting Lower Bound
(CLRS 8.1)

We have seen $\Theta(n \log n)$ and $\Theta(n^2)$ sorting algorithms:(bubblesort), insertion sort, (selection sort), mergesort, quicksort. all have worst-case running time $\Omega(n \lg n)$.

Can we do better than $n \lg n$**?** If we did not find a faster algorithm, it does not mean that there isn't one..

Can we say that it is impossible to sort faster than $\Omega(n \lg n)$ in the worst case? If we could, then this would be what it's called a *lower bound.*

A *lower bound* for a problem is the worst-case running time of the best possible algorithm for that problem.

To prove a lower bound of $\Omega(n \lg n)$ for sorting, we would have to prove that no algorithm, however smart, could possibly be faster, in the worst-case, then $n \lg n$.

So, how to prove a lower bound, without going through all possible algorithms?!?!

To approach this type of problem we have to go back and remind ourselves whhat is an algorithm. The power of an algorithm depends on the primitive instructions we allow, which in turn depend on the machine model.

↓

How fast we can solve a problem depends on the model of computation. The more powerful the machine model, the smaller the lower bound.

(Ultimately, if our model allowed *sort* as a primitive operation, then we could sort in one instruction, hence constant time!!)

Coming back to sorting...

The Comparison Model

- All sorting algorithms that we have seen so far use only comparisons to gain information about the input.

- We will prove that such algorithms have to do $\Omega(n \log n)$ comparisons

- To prove bound, we need *formal model*

  > **Comparison (or Decision tree) Model**
  > - Binary tree where each internal node is labeled $a_i \leq a_j$ (a_i is the i'th input element)
  > - Execution corresponds to root-leaf path
  >   * at each internal node a comparison $a_i \leq a_j$ is performed and branching made
  > - Leaf contains result of computation

- Note: Decision tree model corresponds to algorithms where only comparisons can be used to gain knowledge about input

- Example: Decision tree for sorting 3 elements.
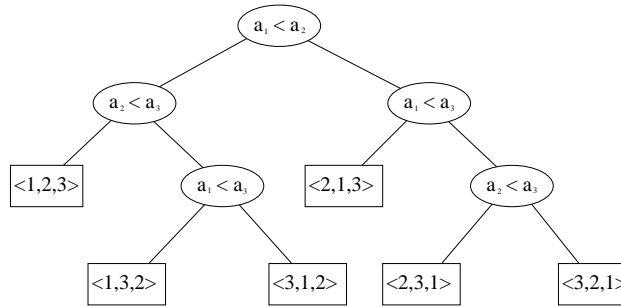
  - a leaf contains permutation giving sorted order.

- Worst case number of comparisons performed corresponds to maximal height of tree ; therefore lower bound on height $\Rightarrow$ lower bound on sorting.

Sorting Lower Bound in the Comparison Model

Theorem: Any decision tree sorting n elements has height $\Omega(n \log n)$

Proof:

  - Assume elements are the (distinct) numbers 1 through n
  - There must be $n!$ leaves (one for each of the $n!$ permutations of n elements)
  - Tree of height h has at most 2^h leaves

$$
\begin{aligned}
2^h \geq n! \Rightarrow h \ &\geq \ \log(n!) \\
&= \ \log(n(n-1)(n-2)\cdots(2)) \\
&= \ \log n + \log(n-1) + \log(n-2) + \cdots + \log 2 \\
&= \ \sum_{i=2}^{n} \log i \\
&= \ \sum_{i=2}^{n/2-1} \log i + \sum_{i=n/2}^{n} \log i \\
&\geq \ 0 + \sum_{i=n/2}^{n} \log \frac{n}{2} \\
&= \ \frac{n}{2} \cdot \log \frac{n}{2} \\
&= \ \Omega(n \log n)
\end{aligned}
$$