Lower Bounds on Sorting
Bound for All Algorithms

S.V. N. (vishy) Vishwanathan

University of California, Santa Cruz
vishy@ucsc.edu

February 3, 2016
Given an array \( \{ a_1, a_2, \ldots, a_n \} \)

Assume all \( a_i \)'s are distinct

Algorithm can only compare \( a_i \) with \( a_j \), that is, it can ask is \( a_i \leq a_j \)?
A Decision Tree

- Full binary tree
- Each node annotated by $i : j$ to indicate that $a_i$ was compared with $a_j$
- Two edges from each node
  - What comparison did the algorithm perform next, if $a_i \leq a_j$?
  - What comparison did the algorithm perform next, if $a_i > a_j$?
- Each leaf of this tree corresponds to a permutation
Insertion Sort

Insertion Sort algorithm is demonstrated for the following sequence: $\langle 1, 2, 3 \rangle$.

The process involves comparing each element with the preceding elements in the sorted portion of the list and moving it to its correct position.

The diagram shows the step-by-step process of sorting:

1. Compare $\langle 1, 2, 3 \rangle$ with $\langle 1, 3, 2 \rangle$, $\langle 3, 1, 2 \rangle$, $\langle 2, 3, 1 \rangle$, and $\langle 3, 2, 1 \rangle$.
2. Insertion into the correct position after each comparison.
3. The final sorted sequence is $\langle 1, 2, 3 \rangle$. 

This process continues until all elements are properly arranged in ascending order.
Any correct sorting algorithm must be able to produce each of the $n!$ permutations on $n$ elements. The decision tree must have at least $n!$ leaves.
Any comparison sort algorithm requires $\Omega(n \log n)$ comparisons in the worst case.
Proof

- Any binary tree of height $h$ has no more than $2^h$ leaves
- We need
  \[ n! \leq 2^h \]
- This implies
  \[ h \geq \log(n!) \]
  \[ = \Omega(n \log n) \]

Note: prove the last line yourself.
Questions?