Australian National University

COMP3600/6466 Algorithms

Lecture 8

S2 2016

Dr. Hassan Hijazi

Prof. Weifa Liang

Comparison-based Sorting

Is the running time of my algorithm the **best possible**?

Let's take **merge-sort** with running time $= \Theta(n \lg n)$

Is it asymptotically **optimal**?

I need to prove that there are **NO** other algorithms in the entire Observable Universe that solves my problem with a running time that is asymptotically better..

Comparison-based Sorting

Let's restrict ourselves to comparison-based sorting algorithms

Algorithms can only compare the value of elements.
Move them around, and apply basic logical operations.

Two equivalent problems:

- **Problem 1.** Sort A in increasing order.
- **Problem 2.** Determine which order the elements of A are in.
  For example, the order of $A = (20, 40, 10, 30, 50)$ is $(2, 4, 1, 3, 5)$.

There are $n!$ possible answers given a n-element list.

Example

Given a 3-element sequence a_1, a_2, and a_3,
sort the sequence in increasing order,
there are $3 = 6$ possible sorted sequences are as follows.

1. $a_1 \leq a_2 \leq a_3$, or its corresponding indices $\langle 1, 2, 3 \rangle$
2. $a_1 \leq a_3 \leq a_2$, or its corresponding indices $\langle 1, 3, 2 \rangle$
3. $a_2 \leq a_1 \leq a_3$, or its corresponding indices $\langle 2, 1, 3 \rangle$
4. $a_2 \leq a_3 \leq a_1$, or its corresponding indices $\langle 2, 3, 1 \rangle$
5. $a_3 \leq a_1 \leq a_2$, or its corresponding indices $\langle 3, 1, 2 \rangle$
6. $a_3 \leq a_2 \leq a_1$, or its corresponding indices $\langle 3, 2, 1 \rangle$

Decision Tree

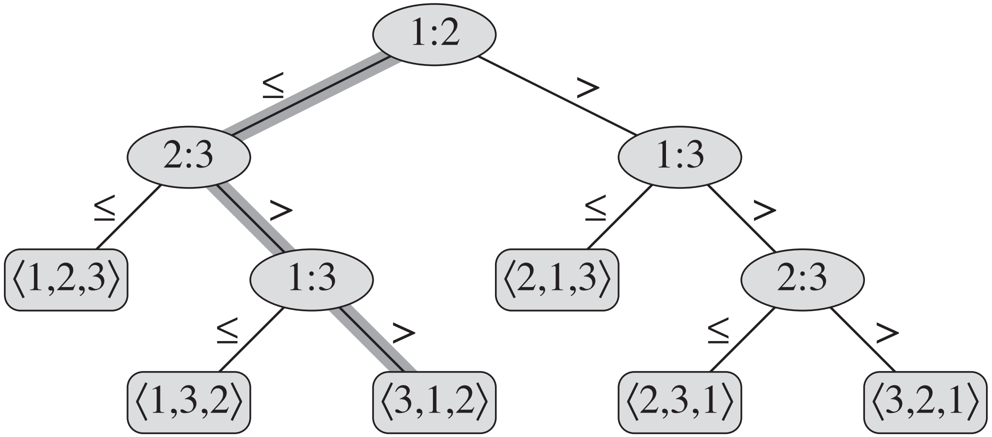
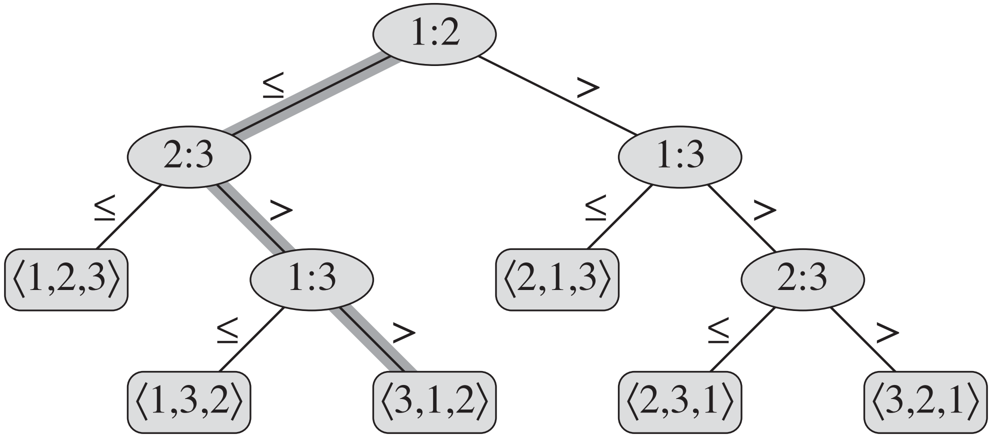


Figure 8.1 The decision tree for insertion sort operating on three elements. An internal node annotated by $i\!:\!j$ indicates a comparison between a_i and a_j. A leaf annotated by the permutation $\langle \pi(1), \pi(2), \ldots, \pi(n) \rangle$ indicates the ordering $a_{\pi(1)} \le a_{\pi(2)} \le \cdots \le a_{\pi(n)}$. The shaded path indicates the decisions made when sorting the input sequence $\langle a_1 = 6, a_2 = 8, a_3 = 5 \rangle$; the permutation $\langle 3, 1, 2 \rangle$ at the leaf indicates that the sorted ordering is $a_3 = 5 \le a_1 = 6 \le a_2 = 8$. There are $3! = 6$ possible permutations of the input elements, and so the decision tree must have at least 6 leaves.

Australian National University

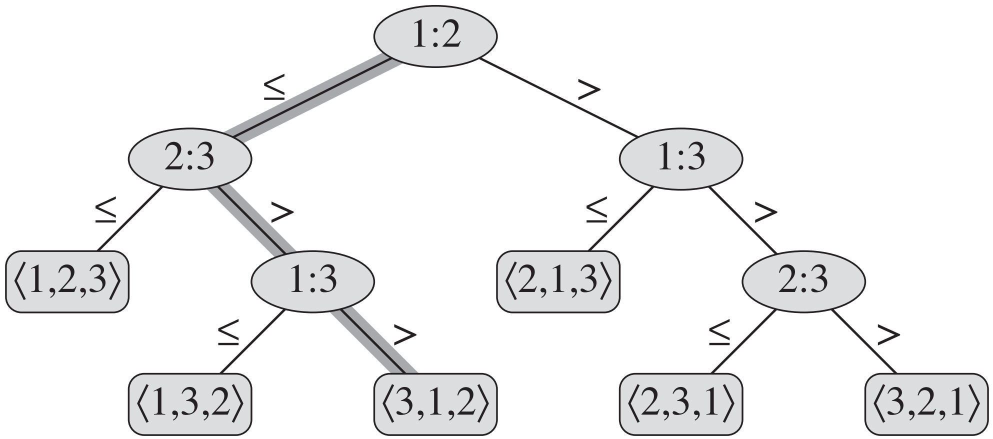
Decision Tree



Important observations:

- The number of nodes from the tree root to a tree leaf corresponds to the number of comparisons to sort a sequence
- There are $n!$ leaves in the binary comparison tree as there are $n!$ different sorted sequences.

Minimising longest path



Minimising the number of comparisons of sorting n elements is equivalent to minimising the depth of the binary comparison tree

How to construct a binary comparison tree that has $n!$ leaves such that the longest path from the root to a leaf is minimised?

Australian National University

Binary Tree Depth

The depth of any binary tree that contains at least 2^h leaves is $\geq h - 1$

Thus, if a binary tree contains $n!$ leaves, its minimum depth is $\lceil \log n! \rceil$

Finally: $\log(n!) = \Theta(n \log n)$ by Stirling's formula.
Stirling's Approximation $n! = \sqrt{2\pi n}(\frac{n}{e})^n(1 + \Theta(\frac{1}{n}))$.