Lower Bound on Sorting

Goals

- Learn definition of decision tree.
- Apply decision-tree concept to comparison-based sorting.
- Can explain why comparison-based sorting always has worst-case time-cost of $\Omega(n \log n)$.

- Reading: Shaffer §7.16 or CLRS §8.1, plus AHU supplement.
Warmup

Q. How many ways are there to arrange \( n \) items in a sequence?

A. ______________.

Each way is called a ___________.

Theorem: a binary tree with height \( h \) has, at most, \( 2^h \) leaves.

Proof: (on whiteboard, by structural induction)
Warmup

• Q. How many ways are there to arrange \( n \) items in a sequence?
• A. \( n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 \)
• Each way is called a permutation.

• Theorem: a binary tree with height \( h \) has, at most, \( 2^h \) leaves.
• Proof: (on whiteboard, by structural induction)
Warmup

• Q. How many ways are there to arrange \( n \) items in a sequence?
• A. \( n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 \)
• Each way is called a permutation.

• Theorem: a binary tree with height \( h \) has, at most, \( 2^h \) leaves.
• Proof: (on whiteboard, by structural induction)
Decision Tree

- Tree, usually binary, representing the multiple outcomes of a series of related decisions.
- Internal nodes represent potential questions (conditions to be tested).
- Links labeled with both possible results.
- Leaves represent conclusions or actions.

Decision tree for financial advice (toy)

Goodrich and Tamassia, *DS&A*
Decision Tree

- Tree, usually binary, representing the multiple outcomes of a series of related decisions.
- Internal nodes represent potential questions (conditions to be tested).
- Links labeled with both possible results.
- Leaves represent conclusions or actions.

Decision Tree

• Tree, usually binary, representing the multiple outcomes of a series of related decisions.

• Internal nodes represent potential questions (conditions to be tested).

• Links labeled with both possible results.

• Leaves represent conclusions or actions.

• Q. Can you think of other examples of (or like) a decision tree?
Decision Tree

- Tree, usually binary, representing the multiple outcomes of a series of related decisions.
- Internal nodes represent potential questions (conditions to be tested).
- Links labeled with both possible results.
- Leaves represent conclusions or actions.

Q. Can you think of other examples of (or like) a decision tree?
- “Choose your own adventure” books
Decision Tree (for sorting)

- Every sorting algorithm must decide how to re-arrange the input sequence into sorted order. (It must diagnose the array's “disease.”)
- If the sort accepts general keys in a total order, then it can only compare two keys at a time.
- The series of potential decisions can be represented in a binary decision tree.
- A single execution corresponds to one path from root to a leaf.
Decision Tree (Insertion sort)

Input: array $A[1..3] =$

\[
\begin{array}{ccc}
\quad a, & b, & c \\
\end{array}
\]


with $a$, $b$, $c$ unknown.

Aho, Hopcroft and Ullman
Decision Tree (sorting)

• Different sorting algorithms will produce different decision trees, based on the order they perform comparisons.

• Every sorting algorithm, given a sequence of size \( n \), must be able to trace a path through a decision tree with ____ leaves, . . .

• . . . because if not, ________.
Decision Tree (sorting)

- Different sorting algorithms will produce different decision trees, based on the order they perform comparisons.
- Every sorting algorithm, given a sequence of size $n$, must be able to trace a path through a decision tree with $n!$ leaves, . . .
- . . . because if not, the algorithm is not correct.
- One leaf for each possible permutation.
Size of a decision tree

• As shown earlier, a binary tree with \( n! \) leaves has height that is at least \( \log n! \).

• Thus, for any comparison-based sorting algorithm, its worst-case input will trace a decision-tree path of length \( \Omega(\log n!) \).

• Each comparison takes at least constant time.

• \((\text{On elmo}): \ (n/2)^{\lceil n/2 \rceil} \leq n! \leq n^n\)

• Therefore, \( \Omega(\log n!) = \Omega(n \log n) \).
Size of a decision tree

• In fact, the *average* of all path-lengths from root to leaf is $\Omega(n \log n)$
  – Proof sketch is in supplemental materials.

• So, no comparison-based sorting algorithm can offer asymptotic time-cost $o(n \log n)$

• Decision tree assumes binary comparisons: occasionally this does not apply (special keys)
Summary

• The series of possible comparisons performed by a sorting algorithm can be organized into a *decision tree*

• Tree must have \( n! \) leaves, thus height
  \[
  h \geq \log n!
  \]

• Worst-case input causes algorithm to execute \( h \) comparisons, which requires \( \Omega(n \log n) \) time

• Often useful as a lower bound for other algorithms (e.g., BST traversal, convex hull)