CSCI 470: Lower Bounds for Sorting, Practice Exam

Vijay Chaudhary

September 21, 2023

Department of Electrical Engineering and Computer Science
Howard University

Overview

1. A quick update
2. Lower bounds of sorting
3. Exam 01

A quick update

Quick Update

- Homework 01 should be graded by Friday.
- You can collect your homework 01 at my office
- Practice exam is ready. We will go over it today.
- Exam 01 will be on Monday, Sep 25, 2023, 9:10 AM EDT

Lower bounds of sorting

Decision Tree Model

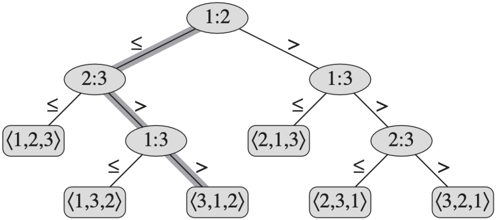

Figure 8.1 The decision tree for insertion sort operating on three elements. An internal node annotated by $i:j$ indicates a comparison between a_i and a_j. A leaf annotated by the permutation $\langle \pi(1), \pi(2), \ldots, \pi(n) \rangle$ indicates the ordering $a_{\pi(1)} \leq a_{\pi(2)} \leq \cdots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

Decision tree model for inseriton sort

- For instance, say, $A = \langle 4, 7, 10 \rangle$, then we first compare $A[1]$ and $A[2]$, which follows the left subtree.
- Then we compare $A[2]$ and $A[3]$, which also satisfies $a_i \leq a_j$, thus leading towards the left subtree.
- Finally, leading to $A[1] \leq A[2] \leq A[3]$, making $\langle 1, 2, 3 \rangle$ as the leaf.

Decision tree model

- If we have an array $A = \langle 6, 8, 5 \rangle$, first $A[1]$ and $A[2]$ is compared, leading to left subtree.
- Then, $A[2]$ and $A[3]$ are compared, where the $a_2 \leq a_3$ fails, therefore, it leads to the right subtree. Here, $A[1]$ and $A[3]$ are compared, which leads to the right subtree again for $a_1 > a_3$.
- The final ordering is $A[3] \leq A[1] \leq A[2]$, making the leaf $\langle 3, 1, 2 \rangle$.

Lower bound

Theorem: Any comparison sort algorithm requires $\Omega(n \lg n)$ comparisons in the worst case.

Lower bound: Proof

Proof Consider a decision tree of height h and l reachable leaves corresponding to a comparison sort on n elements. Because each of the $n!$ permutations of the input appears as some leaf, we have $n! \leq l$. Since a binary tree of height h has no more than 2^h leaves, we have

$$n! \leq l \leq 2^h$$

,

which, by taking logarithms, implies

$$\begin{aligned} h &\geq lg(n!) \\ &= \Omega(n \lg n) \text{ using eq. 3.19 from the book} \end{aligned}$$

Exam 01

Topics/Chapters from CLRS to review for Exam 01

1. Chapter 1; Algorithms a technology
2. Chapter 2; insertion sort, merge sort
3. Chapter 3; O-notation, Ω-notation, Θ-notation
4. Chapter 4; Divide-and-conquer, substitution method for solving recurrences, recurison tree, master method
5. Chapter 6; Heapsort, Priority Queues
6. Chapter 7; Quicksort (upto section 7.2 Performance of quicksort)