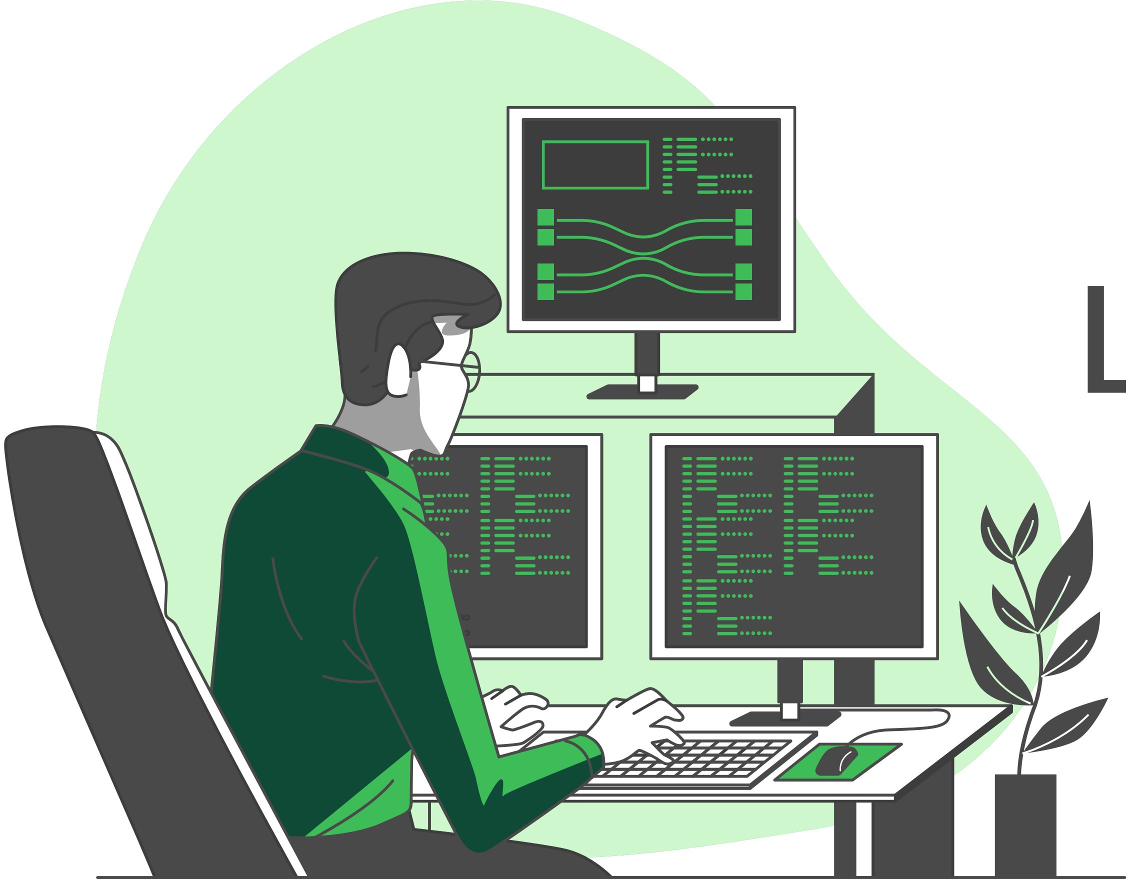
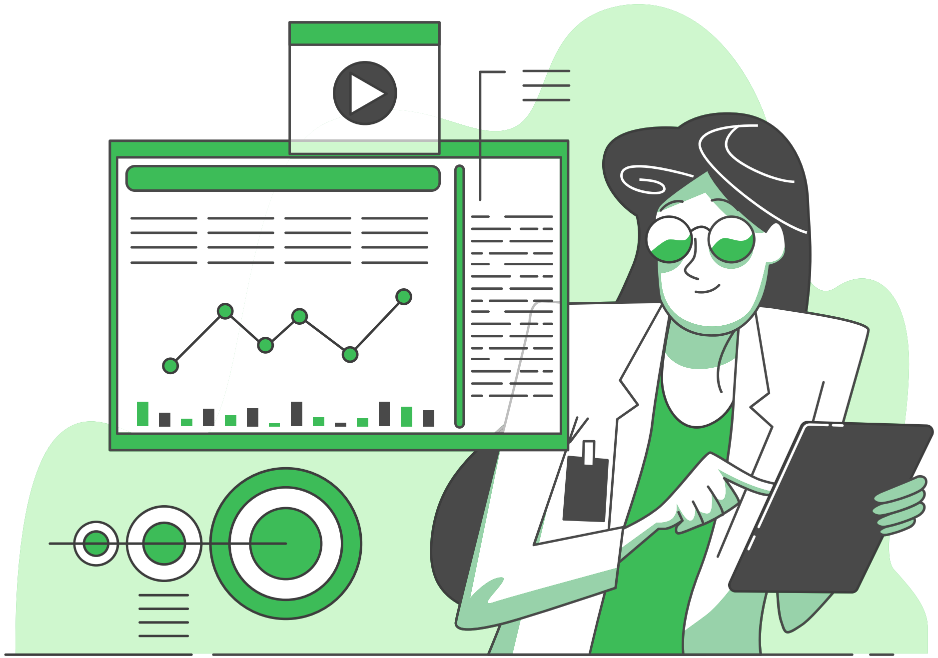


More Linearithmic Sorting

CS 251 - Data Structures and Algorithms

Note:
Slides complement the discussion in class

Table of Contents



01

Better than $n\log(n)$?

Lower bound for comparison-based sorting algorithms

Remember the Sorting Problem

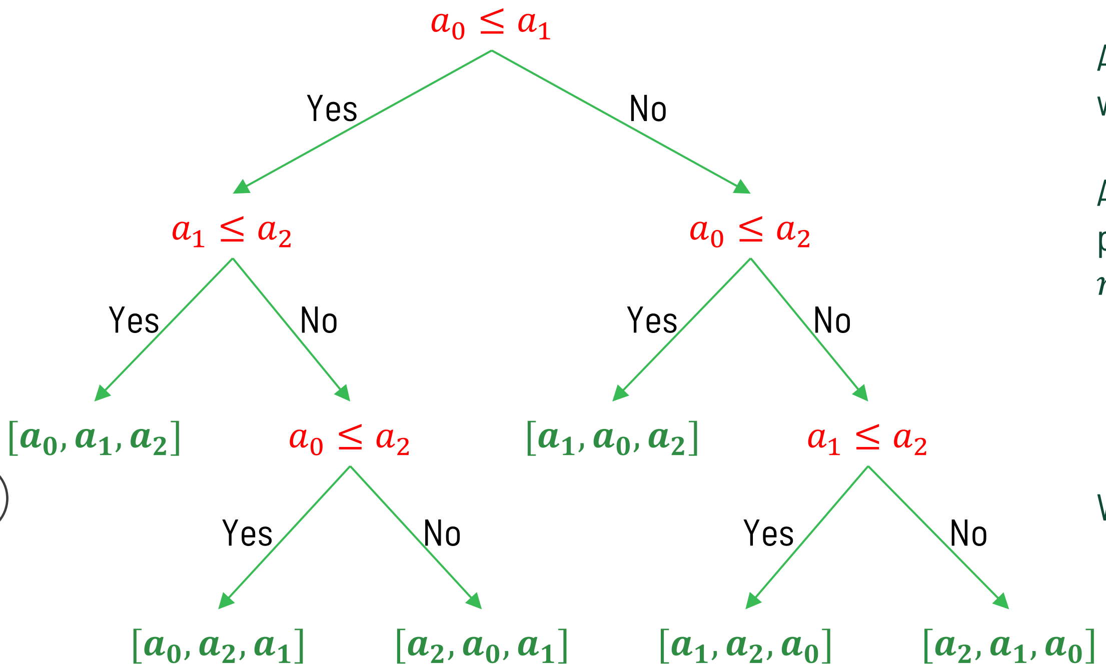
Let $A = [a_0, a_1, a_2, \dots, a_{n-1}]$ be an array with n unsorted elements.

Q: How many permutations of those n elements do we have?
A: $n!$

Q: How many of such permutations correspond to the elements listed in sorted order?
A: At least 1 (remember why?)

Sorting algorithms solve the following problem: ***Given an unsorted array, decide how to permute the array elements such that they are sorted.***

Example: Consider the array $[a_0, a_1, a_2]$. How would a sorting algorithm decide the sorted permutation?



A binary tree has at most 2^h leaves, where h is the height of the tree.

Also, the number of leaves (i.e., permutation of the input array) is $n!$ So,

$$2^h \geq n!$$
$$h \geq \log_2(n!)$$

What does it imply?

Lower Bounds for Comparison-Based Sorting

Theorem: Any comparison sort algorithm requires $\Omega(n\log_2(n))$ comparisons in the worst case.

Proof: We know that $h \geq \log_2(n!)$ from the decision tree model.

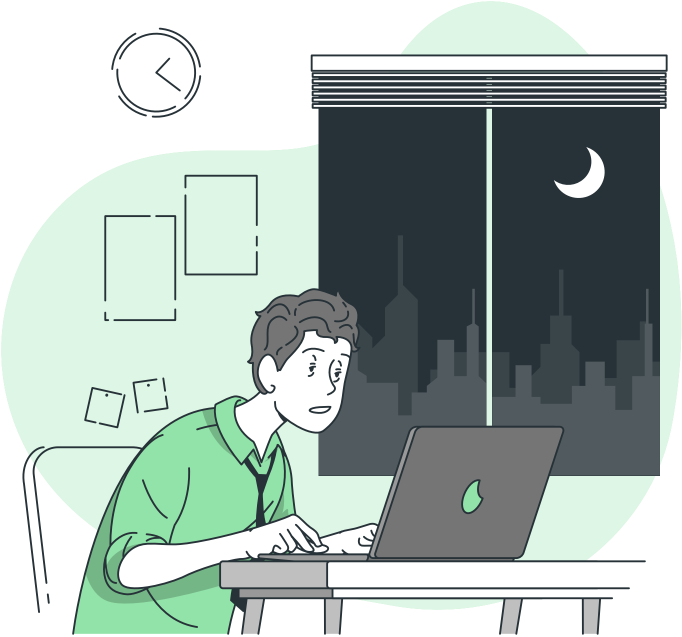
$$\log_2(n!) = \log_2(1 \cdot 2 \cdot 3 \cdot \dots \cdot n)$$
$$= \log_2(1) + \dots + \log_2\left(\frac{n}{2}\right) + \dots + \log_2(n-1) + \log_2(n)$$
$$\geq \log_2\left(\frac{n}{2}\right) + \dots + \log_2(n-1) + \log_2(n)$$
$$= \log_2\left(\frac{n}{2}\right) + \log_2\left(\frac{n}{2}+1\right) + \dots + \log_2(n-1) + \log_2(n)$$
$$\geq \log_2\left(\frac{n}{2}\right) + \log_2\left(\frac{n}{2}\right) + \dots + \log_2\left(\frac{n}{2}\right)$$
$$= \frac{n}{2}\log_2(n)$$

So, $\log_2(n!) \in \Omega(n\log_2(n))$

Since $h \geq \log_2(n!)$, then $h \in \Omega(n\log_2(n))$

Theorem statement from Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. Introduction to Algorithms (The MIT Press)(p. 193).

So, no Faster Sorting?



Nope!

$h \geq n \log(n)$. So, $h \in \Omega(n \log(n))$

Unless...

... we stop sorting by comparing items with each other.

Done!

Do you have any questions?

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik and illustrations by Stories