Can We Sort Faster Than $O\,(n \log n)$?

Can We Sort Faster Than $O(n \log n)$?

- Why ask?
- Maybe our human race is not smart enough yet?
- In the 19th century, John Dalton, through his work on stoichiometry, concluded that each chemical element was composed of a single, unique type of particle. Dalton and his contemporaries believed those were the fundamental particles of nature and thus named them atoms, after the Greek word *atomos*, meaning “indivisible” or “uncut”
- But luckily, we can prove that!

However

- There are some sorting algorithm that is REALLY faster than O(n log n), e.g. Radix sort
- But, those faster algorithms are on a “different basis”
- What we introduced so far, all the algorithms are ***comparison*** sorting

Comparison Sorting

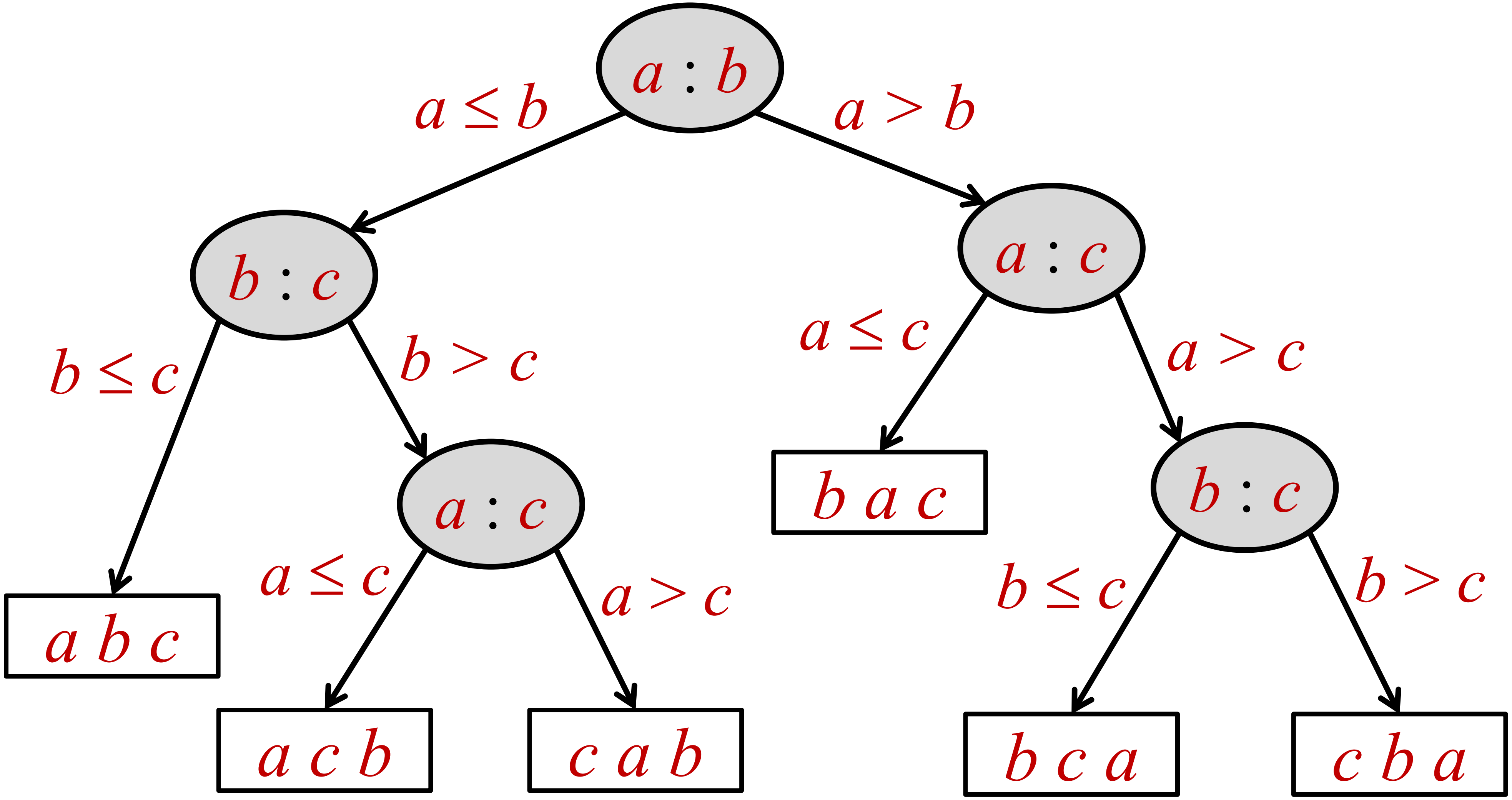
- We say that a sorting algorithm is a ***comparison sort*** if only comparisons between two elements are used to determine the order of the elements.
- What is comparison:
  - `if (a < b) then …`
  - `if (a > b) then …`
  - `if (a == b) then …`

Comparison Sorting

- We say that a sorting algorithm is a *comparison sort* if only comparisons between two elements are used to determine the order of the elements.
- Claim

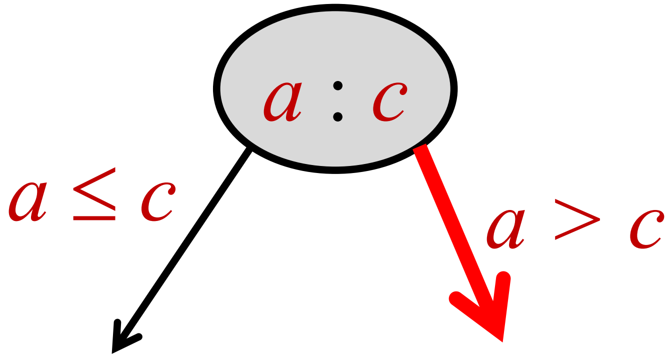
Comparison sort cannot be faster than O(n log n)

Consider Sorting {a,b,c}

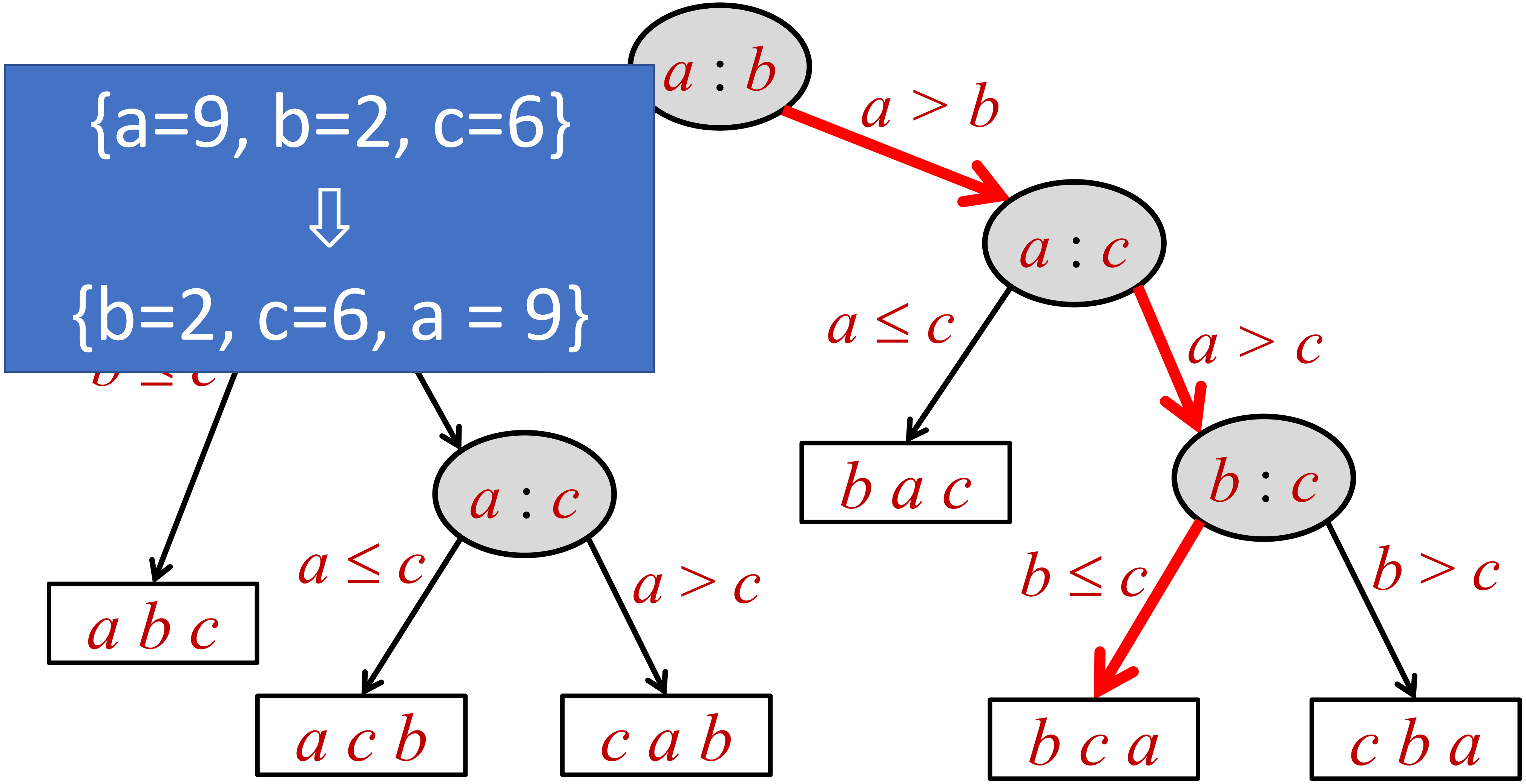


For each node

- Each node is a comparison to determine what to do
  - E.g. Swapping two elements
- How many steps from the root to *any* leave is the number of steps of sorting

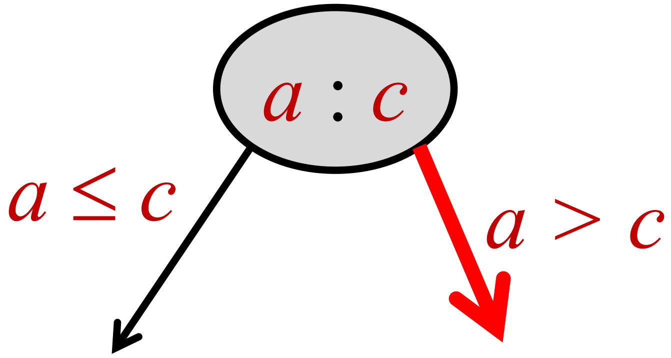


Example: Consider Sorting {a=9,b=2,c=6}

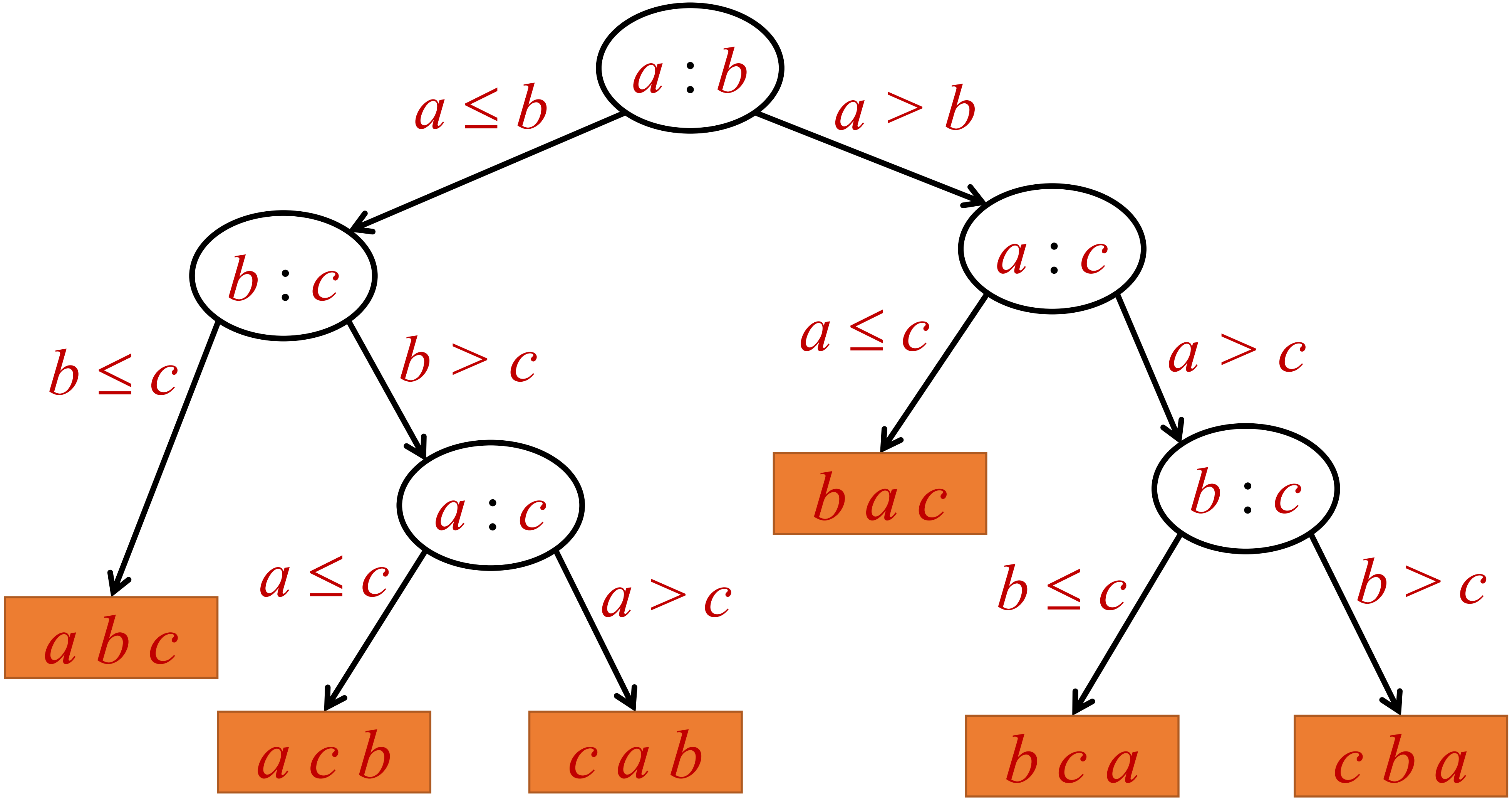


For each node

- Each node is a comparison to determine what to do
  - E.g. Swapping two elements
- How many steps from the root to *__any__* leave is the number of steps of sorting
- But, we can have a tree that some path is very long and some very short? (Imbalanced tree)
- So the "average" will be the height of a totally balanced tree!
  - = when every leaf takes the same #steps to the root



For 3 Items, How many leaves?



How many leaves?

- For 3 items in original order of {a,b,c}, we can sort it into any permutation

abc, acb, bac, bca, cab, cba

- For 5 items?
  - 120 permutations!
- For n items?
  - $n!$
- If we have $n!$ leaves, what is the height of the balanced tree?

Simple Math

- Height 1 has 2 leaves
- Height 2 has 4 leaves
- Height 3 has 8 leaves
- Height k has 2^k leaves
- Given $n!$ leaves, the height is $\log(n!)$

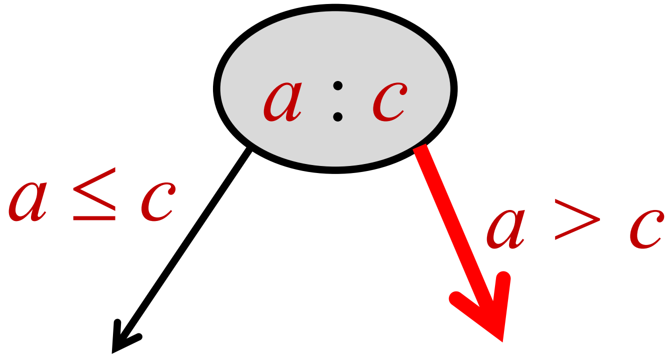
Stirling's Approximation

$$n! \approx \sqrt{2\pi \cdot n}\left(\frac{n}{e}\right)^n > \left(\frac{n}{e}\right)^n$$

- Height of the tree of $n!$ leaves:

$$\log(n!) > n \log(n/e) = \Omega(n \log n)$$

For each node



- Each node is a comparison to determine what to do
  - E.g. Swapping two elements
- How many steps from the root to *any* leave is the number of steps of sorting

$$\Omega\ (n \log n)$$

- Conclusion, comparison sorting takes at least Ω $(n \log n)$ steps/comparisons!

To $\Omega(n \log n)$ and beyond

- Will there be any sorting that is faster than $\Omega(n \log n)$
- Then they must not be comparison sort!
  - Radix sort, counting sort, etc.