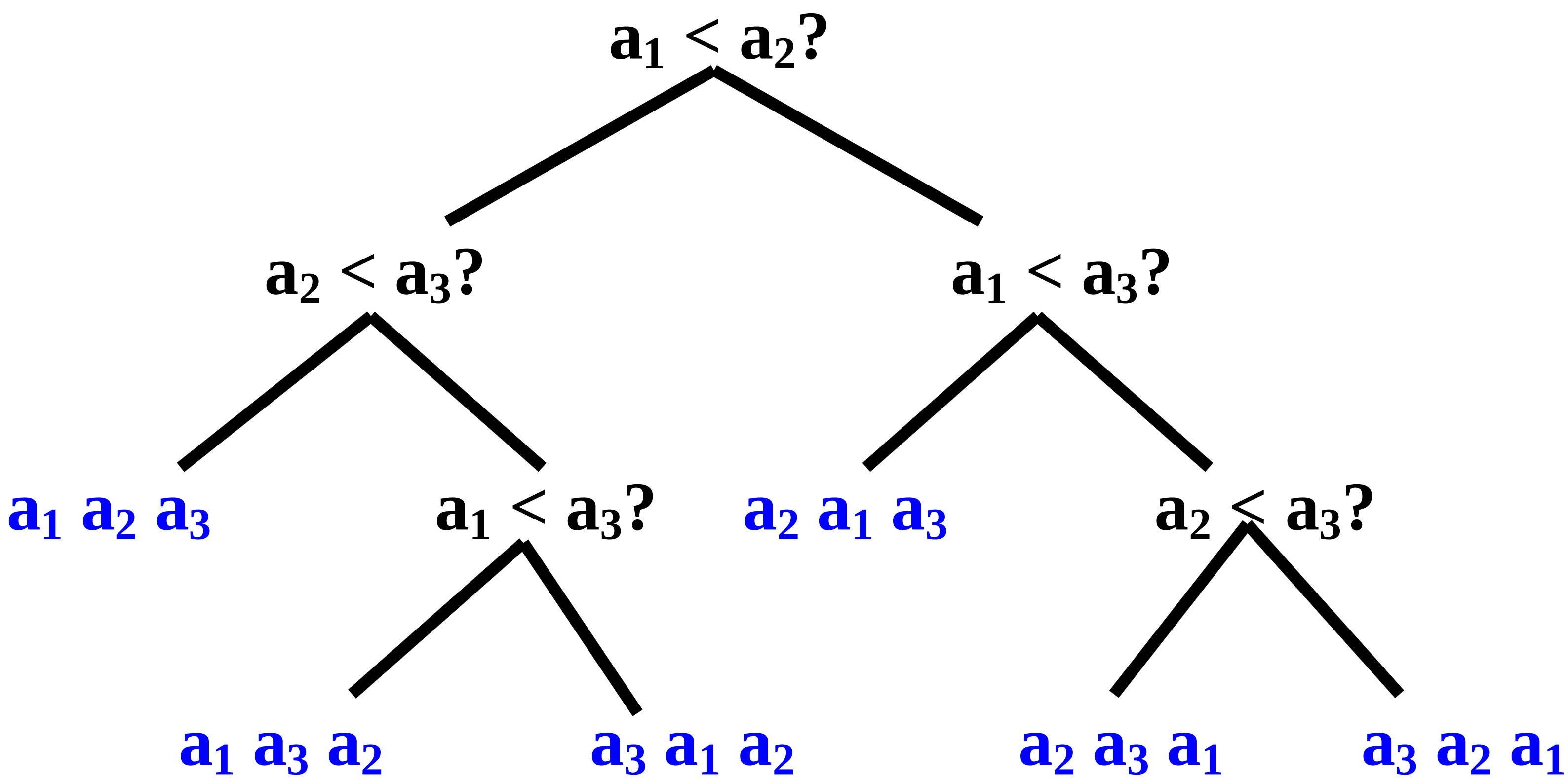
How Fast Can We Sort?

A *decision tree* is a binary tree in which each non-leaf represents a comparison between two elements and each leaf represents a sorted sequence of those elements.

Left branch: Yes

Right branch: No

Example: Apply insertion sort to a1, a2, a3.



A decision tree has one leaf for each permutation of the *n* elements to be sorted.

The number of permutations of *n* distinct elements is ?

$n!$

So a decision tree to sort *n* elements must have *n*! leaves.

The binary-tree theorem:

For any non-empty binary tree t:

1. $$\text{leaves}(t) \le \frac{n(t) + 1}{2.0}$$

2. $$\frac{n(t) + 1}{2.0} \le 2^{\text{height}(t)}$$

By the binary tree theorem, for any non-empty tree t,

$$\mathbf{leaves(t) <= 2^{height(t)}}$$

Since $n! = leaves(t)$, we must have

$$n! \le 2^{height(t)}$$

which implies that

$$\log_2 (n!) \le height(t)$$

In the context of a decision tree, height(t) represents the maximum number of comparisons needed to sort the *n* elements.

So $\log_2(n!)$ <= the maximum number of comparisons to sort n elements.

Therefore,

worstTime(n) >= $\log_2(n!)$

$$\log_2(n!) >= n/2 \log_2(n/2)$$

So

$$\text{worstTime}(n) >= n/2 \log_2(n/2)$$

So worstTime(n) is $\Omega(n \log n)$ for any comparison-based sort.

What can we say about averageTime(n)?

averageTime(n) >= average number of comparisons

= total number of comparisons / n!

In a decision tree, what is the total number of comparisons equal to?

averageTime(n) >= average number of comparisons

= total number of comparisons / n!

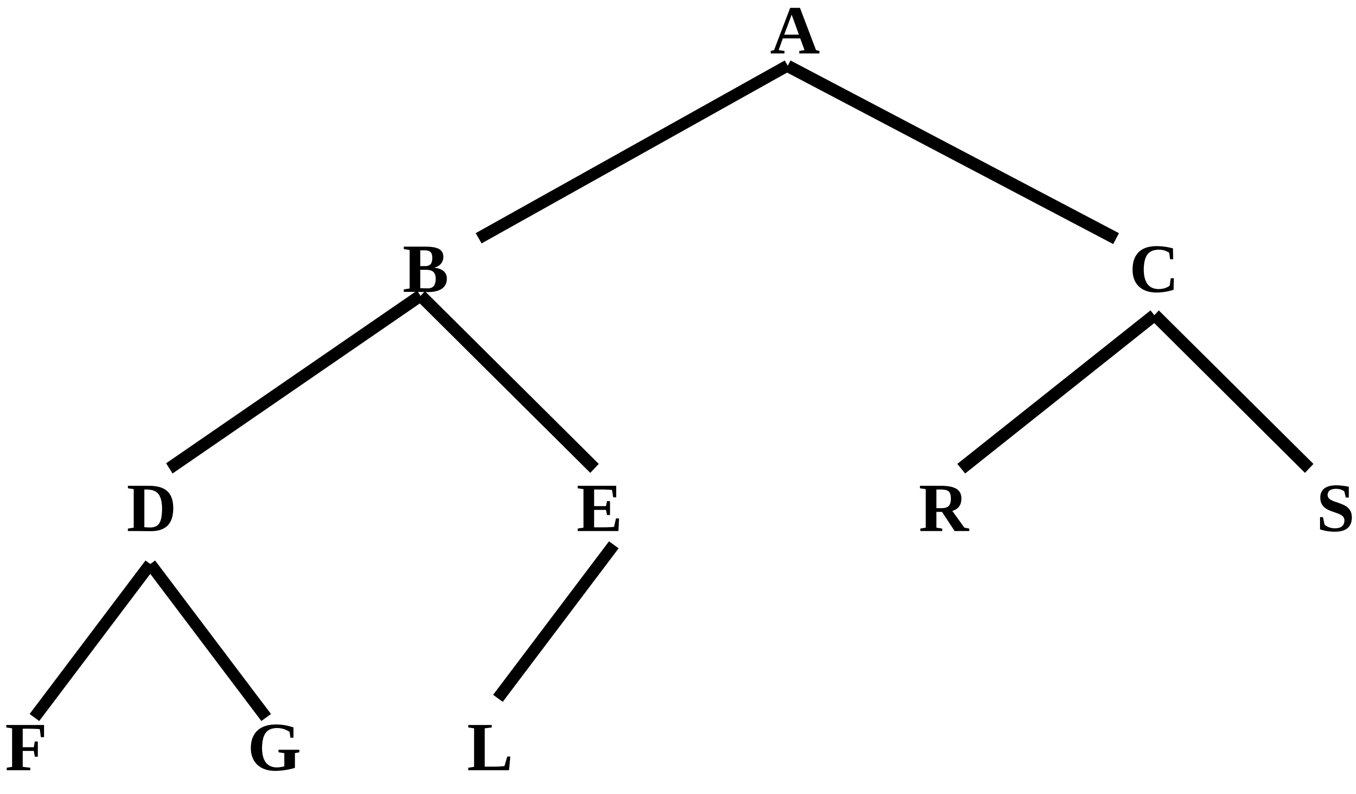
In a decision tree, what is the total number of comparisons equal to?

Hint: The length of each path from the root to a leaf equals the number of comparisons in that path.

The total number of comparisons is equal to the sum of all root-to-leaf path lengths.

Let t be a non-empty binary tree. *E(t)*, the external path length of t, is the sum of the depths of all leaves in t.

For example, find the external path length of the following binary tree:



$$E(t) = 3 + 3 + 3 + 2 + 2 = 13$$

The external path length theorem:

Let t be a binary tree with $k > 0$ leaves. Then

$E(t) >= (k / 2)$ floor $(\log_2 k)$.

E(t), the external path length of tree t, is the sum of all root-to-leaf path lengths in t. So the average number of comparisons is

E(t) / n!

In a decision tree, the number of leaves is *n*!.
so, by the external path length theorem,

averageTime(n) >= average # comparisons

$$= E(t) / n!$$
$$>= (n! / 2) \text{ floor } (\log_2(n!)) / n!$$
$$= (1 / 2) \text{ floor } (\log_2(n!))$$
$$>= (1 / 4) (\log_2(n!))$$
$$>= (n / 8) (\log_2(n / 2))$$

For any comparison-based sort, averageTime(n) is $\Omega(n \log n)$.