9.1-3 Show that there is no sorting algorithm which sorts at least \((1/2^n) \times n!\) instances in \(O(n)\) time.

Think of the decision tree which can do this. What is the shortest tree with \((1/2^n) \times n!\) leaves?

\[
h > \lg(n!/2^n) = \lg(n!) - \lg(2^n) = \Theta(n \lg n) - n = \Theta(n \lg n)
\]

Moral: there cannot be too many good cases for any sorting algorithm!
9.1-4 Show that the $\Omega(n \lg n)$ lower bound for sorting still holds with ternary comparisons.

The maximum number of leaves in a tree of height $h$ is $3^h$, 

$$\lg_3(n!) = \Theta(n \lg n)$$

So it goes for any constant base.