**Theorem 8.1.** For any comparison-based sorting algorithm, the worst-case number of comparisons is $\Omega(n \lg n)$.

**Proof.** For sequences of size $n$, there are $n!$ permutations, each of which are possible outcomes. Consider the decision tree where each node is a comparison between two array positions.

Let $\ell$ be the number of leaves and $h$ the height of the tree. And so

\[
\begin{align*}
    n! & \leq \ell & \text{since every permutation must be a leaf} \\
    \ell & \leq 2^h & \text{since a tree can’t have more than } 2^h \text{ leaves} \\
    h & \leq \lg n! \\
    & = \Theta(n \lg n) \text{ by eq 3.19 in CLRS}
\end{align*}
\]

Hence $h = \Omega(n \lg n)$, and thus there must be a permutation reachable by no less than $\Omega(n \lg n)$ comparisons. □
8.1-3.a. Can a comparison-based sorting algorithm have linear running time for at least half the inputs of size $n$?

Suppose so, that is, suppose there exists a $c$ such that for $\frac{n!}{2}$ of the items, their path is fewer than $cn$ links. This means that in the portion of the tree less than $cn$ links from the root, there are $\frac{n!}{2}$ leaves. In fact, the most possible leaves are $2^{cn}$. Thus,

\[
\frac{n!}{2} \leq 2^{cn} \quad \text{and} \quad \lg(n!) \leq cn + 1
\]

\[
n! \leq 2^{cn+1}
\]

Since $\lg(n!) = \Omega(n \lg n)$, there exists a $d$ such that $\lg(n!) \geq dn \lg n$.

\[
c \geq \frac{\lg(n!)}{n} - \frac{1}{n} \geq \frac{dn \lg n}{n} - \frac{1}{n} = d \lg n - \frac{1}{n}
\]

$\frac{1}{n}$ approaches 0 and $d \lg n$ approaches $\infty$ (slowly). So, $c$ cannot be a constant.

Alternately, we could observe that $\frac{n!}{2} \leq 2^h$, and so

\[
h \geq \lg n! - 1 = \Omega(n \lg n)
\]
8.1-3.b. Can a comparison-based sorting algorithm have linear running time for \( \frac{1}{n} \) of the inputs of size \( n \)?

Suppose so. Then

\[
\frac{n!}{n} \leq 2^{cn}
\]

\[
\lg(n!) - \lg n \leq cn
\]

\[
c \geq \frac{\lg(n!) - \lg n}{n} \geq \frac{dn \lg n}{n} \geq d \lg n - \frac{\lg n}{n}
\]

Since the \( \frac{\lg n}{n} \) term approaches 0, the last expression is increasing. Hence \( c \) is not constant.

Alternately, \( \frac{n!}{n} \leq 2^h \), so

\[
h \geq \lg n! - \lg n = \Omega(n \lg n)
\]
Can a comparison-based sorting algorithm have linear running time for $\frac{1}{2^n}$ of the inputs of size $n$?

Suppose so. Then

$$\frac{n!}{2^n} \leq 2^{cn}$$

$$n! \leq 2^{(c+1)n}$$

$$\lg(n!) \leq (c+1)n$$

$$c \geq \frac{\lg(n!)}{n} - 1$$

$$\geq \frac{dn \lg n}{n} - 1$$

$$= d \lg n - 1$$

Alternately, $\frac{n!}{2^n} \leq 2^h$, so

$$h \geq \lg n! - n = \Omega(n \lg n)$$
8.1-4. The number of permutations is \( k! \cdot k! \ldots k! \), that is, \( (k!)^{n/k} \).

For a decision tree of height \( h \), \( (k!)^{n/k} \leq 2^h \). So,

\[
    h \geq \log((k!)^{n/k})
\]

\[
    = \frac{n}{k} \log(k!)
\]

\[
    = \frac{n}{k} d k \log k \quad \text{for some } d
\]

\[
    = d n \log k
\]

Hence \( \Omega(n \log k) \).