Ryan Pollack

1. We can use a simple method by defining a sequence with a factorial ($n!$) and polynomial (a^n). So:

$$f(n) = \frac{n!}{a^n}$$

Now:

$$\frac{f(n+1)}{f(n)} = \frac{(n+1)!}{a^{n+1}} \cdot \frac{a^n}{n!} = \frac{n+1}{a}$$

When $n \to \infty$, the function goes to infinity as well:

$$n \to \infty \text{ means } \frac{f(n+1)}{f(n)} \to \infty$$

So, the original function $\frac{n!}{a^n}$ is a diverging sequence. Therefore, $n!$ grows faster than a^n, meaning $n!$ dominates any polynomial (or $n^\alpha = O(n!)$) for any real α.

2. We know that $n! = n(n-1)(n-2)\ldots(2)(1)$

$$= n \cdot n(1-\tfrac{1}{n}) \cdot n(1-\tfrac{2}{n}) \ldots n(\tfrac{2}{n}) \cdot n(\tfrac{1}{n})$$

$$= n^n(1-\tfrac{1}{n})(1-\tfrac{2}{n})\ldots(\tfrac{2}{n})(\tfrac{1}{n})$$

The dominating factor is n^n.

For $\alpha > 0$,

$n > \alpha$

$n^n > \alpha^n$

$\alpha^n < n^n \approx n!$

$\frac{\alpha^n}{n!} < 1$

Since the function is less than 1 and therefore a fraction, $n!$ grows faster than α^n. Thus, $n!$ dominates any exponential (or $\alpha^n = O(n!)$) for any real α.

3. For $n=1$, $1! = 1 = 1^1 = 1$

So for $n=1$, $n! \leq n^n$ is true.

Now, let's say that $n=k$, where:

$k! \leq k^k$

To show that it's true for $n=k+1$, we want to show:

$(k+1)! = (k+1)k!$

So $k! \leq k^k$

$\rightarrow (k+1)k! \leq k^k(k+1) \leq (k+1)^k(k+1)$

$\rightarrow (k+1)k! \leq (k+1)^{k+1}$

$\rightarrow (k+1)! \leq (k+1)^{k+1}$

So, it is also true when $n=k+1$.

Therefore, by method of induction, $n! \leq n^n$ for $n \geq 1$.

4. We can do a very similar thing as problem 3.

For $n=2$ $\qquad$ LHS $= 2! = 2$ $\quad$ RHS $= \left(\frac{2}{2}\right)^{\left(\frac{2}{2}\right)} = 1^1 = 1$

$2 \geq 1$ so it holds true for $n=2$

Now, let's say that $n=k$, where:

$k! \geq \left(\frac{k}{2}\right)^{\left(\frac{k}{2}\right)}$

To show that it's true for $n=k+1$, we want to show:

$(k+1)! \geq \left(\frac{k+1}{2}\right)^{\left(\frac{k+1}{2}\right)}$

So $k! \geq \left(\frac{k}{2}\right)^{\left(\frac{k}{2}\right)}$

$(k+1)k! \geq \left(\frac{k}{2}\right)^{\left(\frac{k}{2}\right)}(k+1)$

$\log(k+1)! \geq \log\left(\left(\frac{k}{2}\right)^{\left(\frac{k}{2}\right)}(k+1)\right)$

$\log(k+1)! \geq \left(\frac{k}{2}\right)\log\left(\frac{k}{2}\right) + \log(k+1)$

$\log(k+1)! \geq \left(\frac{k}{2}\right)\log\left(\frac{k+1}{2}\right) + \frac{1}{2}\log\left(\frac{k+1}{2}\right)$

$\log(k+1)! \geq \left(\frac{k}{2}+\frac{1}{2}\right)\log\left(\frac{k+1}{2}\right) \longrightarrow \left(\frac{k+1}{2}\right)\log\left(\frac{k+1}{2}\right) \longrightarrow \log\left(\frac{k+1}{2}\right)^{\left(\frac{k+1}{2}\right)}$

$\log(k+1)! \geq \log\left(\frac{k+1}{2}\right)^{\left(\frac{k+1}{2}\right)}$

$(k+1)! \geq \left(\frac{k+1}{2}\right)^{\left(\frac{k+1}{2}\right)}$

So, it is also true when $n=k+1$.

Therefore, by method of induction, $n! \geq \left(\frac{n}{2}\right)^{\left(\frac{n}{2}\right)}$ for $n \geq 2$

5. To find this, we can get the upper and lower bound of $\ln(n!)$ and use Squeeze Theorem.
We start with what we know:

$$\ln(n!) = \ln(1) + \ln(2) + \dots \ln(n-1) + \ln(n)$$

For upper bound:

$$\ln(1) + \ln(2) + \dots + \ln(n) \le \ln(n) + \ln(n) + \dots + \ln(n)$$
$$'' \qquad '' \le n \cdot \ln(n)$$

For lower bound:

$$\ln(1) + \dots + \ln(\tfrac{n}{2}) + \dots + \ln(n) \ge \ln(\tfrac{n}{2}) + \dots + \ln(n)$$
$$'' \qquad '' \ge \ln(\tfrac{n}{2}) + \ln(\tfrac{n}{2}+1) + \dots + \ln(n-1) + \ln(n)$$
$$'' \qquad '' \ge \ln(\tfrac{n}{2}) + \dots + \ln(\tfrac{n}{2})$$
$$'' \qquad '' \ge \tfrac{n}{2} \cdot \ln(\tfrac{n}{2})$$
$$'' \qquad '' \ge n \cdot \ln(n)$$

So, $\hat{n}\ln(n) \le \ln(n!) \le n\ln(n)$
where the upper and lower bounds are equivalent, meaning the function of $\ln(n!)$ inbetween must also be equivalent. Therefore, by Squeeze Theorem, $\ln(n!) = \Theta(n\ln(n))$.

1. The total possible ways to list 'n' values is $n!$ because there are 'n' distinct elements:

$$n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 = n!$$

2. Putting out the order permutation of the list, which we'll recognize as "$i_1, i_2, \dots, i_n$", which implies "$a_{i_1} \le a_{i_2} \le \dots \le a_{i_n}$".
So, given the order permutation, just iterate over it from left to right and print in order. The output is the sorted list which takes linear time as each index gets accessed once.
If the steps to sort are known, first apply the steps to sort and let the after-sort list be in form "$a_{i_1}, a_{i_2}, \dots, a_{i_n}$".
Then order permutation is "$i_1, i_2, \dots, i_n$" by definition of order permutation.

3. If there isn't an ordering permutation, then we can't sort the list without any comparison tests. Sorting can only be as easy as finding the order permutation because one can be used to do the other.

4. The amount of information one comparison encode is one. (If i comparisons are performed, i bits of information are encoded).
The amount of possibilities for the order permutations is $n!$, thus it must have at least $\log_2(n!)$ many bits to encode order permutations.
The amount of information comparisons need to encode for sorting must be $\geq$ the information required to encode an order permutation because finding order permutations is as hard as sorting. Therefore, $i \geq \log_2(n!)$, so you must perform at least approximately $\log_2(n!)$ many comparisons.

5. Using Sterling's approximation, we know $\log_2(n!) = \Theta(n\log n)$. So, any sorting algorithm must take time $\Theta(n\log n)$ as the worst case. The merge sort also takes time $\Theta(n\log n)$ as the worst case. Therefore, mergesort is, asymptotically speaking, equal to or more efficient than any other sorting algorithm.

Bonus $\ln(n!)$
For tightest upper bound, we already know $\ln(n!) = O(n\ln(n))$.
$\Theta(\log_e n^n) = \Theta(n\log_e n) = \Theta(n\ln_e n) = \Theta(n\ln(n))$
Therefore, the tightest bound is $\Theta(n\log(n))$ time.