1. , X is antisymmetric

X is transitive

1. X has totality (implies reflexivity)

1)

Assume X is a finite element set with n elements.

If n = 1, any arbitrary element is totally ordered on X.

If n > 1, we have

If then a is the minimum element in X

If then which means b is the minimum element

If we remove the minimum element **m** from set **X** recursively we will find all of the permutations within X. Since the total ordering is merely a permutation, and there exists n! permutations in an n-element set, we can have **n! different total ordered permutations.**

2)

If the total ordering permutation of a set is, for example: 7, 6, 8 this would tell us that elements because the permutation represents the elements in sorted order. Since the elements are already sorted we do not have to commit any additional comparisons or tests.

3)

Without having the ability to compare and only given a partial/incomplete total ordering permutation you cannot determine the order of the entire set. The remaining elements if arbitrarily placed would not be certainly sorted we would be left with uncertainty in regards to the order of the set.

4)

To sort n distinct numbers, a sorting algorithm must distinguish between n! different permutations. A correct algorithm takes a set of values as an input and produces some value or set of values as output. To solve a sorting problem the algorithm takes an instance of the set and outputs a permutation of the input sequence such that:

(a total order). In order to construct a total order, there must be n amount of information inside the total order. While algorithms may work more than the necessary amount to generate a total order, the lower bound of information is only required.

5)

Because there is only a yes/no decision being made to distinguish between n! permutations the amount of tests you need is: because you’re only given 1-bit of information from each comparison.

6)

**Direct Proof:**

Because: , that shows proof of our statement.

**Proof by Induction:**

Base Case:

Inductive Hypothesis:

Inductive Conclusion:

Inductive Step:

Binomial Theorem :

Base Case + Inductive Step is true.

is true for all

7)

8)

is the operational lower bound for all comparison based algorithms and the operational upper bound for some. The minimal number of test/comparisons needed to sort a list is equal to the amount of information necessary to sort the list. Since comparisons only give 1 bit of information you will need to do at least many operations.

9)

Merge sort has a lower bound time complexity of O(n ln(n)) while also having an upper bound of O(n ln(n)), making it more efficient than any other sorting algorithm.

10)

Tightest bound on n!