Computer Science 385
Design and Analysis of Algorithms
Siena University
Spring 2026

# Establishing Lower Bounds Practice

---

## Lower Bounds

A *lower bound* gives

An established lower bound means

Lower bounds can be exact but are often expressed as a Big-______ efficiency class.

A lower bound is *tight*

### Lower Bound Examples

- the number of comparisons needed to find the largest element in an unordered set of $n$ numbers

  Lower bound: $\Omega(\qquad)$ tight?

- number of comparisons needed to sort an arbitrary array of size $n$

  Lower bound: $\Omega(\qquad)$ tight?

- number of comparisons necessary for searching in a sorted array of $n$ numbers

  Lower bound: $\Omega(\qquad)$ tight?

- the number of comparisons needed to determine if all elements of an array of $n$ elements are unique

  Lower bound: $\Omega(\qquad)$ tight?

- number of steps needed to multiply two $n$-digit integers
  Lower bound: $\Omega($ $)$ tight?
- number of multiplications needed to multiply two $n \times n$ matrices
  Lower bound: $\Omega($ $)$ tight?

---

**Trivial Lower Bounds**

A *trivial lower bound* is established by

**Generating all permutations of a set of $n$ items**

Trivial lower bound: $\Omega($ $)$ Tight bound?

Why?

**Evaluating a polynomial of degree $n$, $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$**

Trivial lower bound: $\Omega($ $)$ Tight bound?

Why?

**Multiply two $n \times n$ matrices**

Trivial lower bound: $\Omega($ $)$ Tight bound?

Why?

**Traveling Salesman with $n$ cities**

Trivial lower bound: $\Omega($ $)$ Tight bound?

Why?

**What about finding an element in a collection of size $n$?**

---

## Information Theoretic Lower Bounds

An *information-theoretic lower bound* is based on

**Binary search in a sorted array of size $n$**

**A Decision Tree: Find minimum of three numbers $a, b, c$**

**Decision tree properties**

- The number of leaves must be

- The operation of an algorithm on a particular input is modeled by

- The number of comparisons is equal to

- Worst-case behavior is determined by

Any such tree with a total of $l$ leaves (outcomes) must have $h \geq$

## Decision Tree for Selection Sort of 3 elements

## Decision Tree for Insertion Sort of 3 elements

*Any* **comparison-based** sorting algorithm can be represented by a decision tree. Assuming our input has $n$ values:

- The number of leaves (outcomes) must be $\geq$

- The height of binary tree with ____ leaves $\geq$

- This tells us the number of comparisons in the worst case
  $C_{worst}(n) \geq$

  for **any** comparison-based sorting algorithm. (!!)

- Is this bound tight?

The idea of *adversary arguments* to find a lower bound is discussed in the notes.