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

In-Class Introductory Topics

What is an *algorithm*?

The notion of an *algorithm* is much older than modern computing. For purposes of computers, an algorithm needs to be:

We will explore other properties of algorithms by example.

Example Algorithm

What does this function/method compute?

```
int max(int a, int b, int c) {
  if (a > b) {
    if (a > c) return a;
    else return c;
  }
  else {
    if (b > c) return b;
    else return c;
  }
}
```

The algorithm has three *inputs* (_____)

and one *output* (_____)

The algorithm is defined *precisely* and is *deterministic*.

Determinism is a key feature:

A *non-deterministic* procedure:

How can we introduce non-determinism in an algorithm?

Important properties:

finiteness:

correctness:

generality:

efficiency:

How do we know that our example algorithm terminates?

How can we *verify* correctness?

A variant that is not general:

```
int max(int a, int b, int c) {
   if (a > 10 && b < 10 && c < 0) return a;
}
```

Does it give any incorrect answers?

Give a set of inputs for which it does not produce any answer.