Mapping Integers to the Smallest Description of an Integer Possible in Python

John Seibert
Advisor: William Calhoun
Bloomsburg University of Pennsylvania

Background and Motivation

- **The Halting Problem**: If we are given an arbitrary computer program and an input, **will a program always give an output?**
- Cannot say "yes" to above due to the following arguments:
  - Limited size of computer memory
  - Possibility of program non-termination during iteration/loop
  - Program execution time could be longer than human lifespan
  - Programs that take itself as input (proposed by Cantor)
- **Project Goal:**
  - Construct a mathematical function in Python mapping integers to smallest integer based on a set of algorithms

Original Cost Function

- **Attempt at computability**: let m be an input and C(m) its output
- Consider the original definition of the cost of an integer, m:
  - Given two operands, a and b, such that if a + b = m, or a * b = m, the goal is to **minimize C(a) + C(b)**
  - The current cost, C(m), starts with the input and follows the chart
  - Easily computable using smaller values of the function

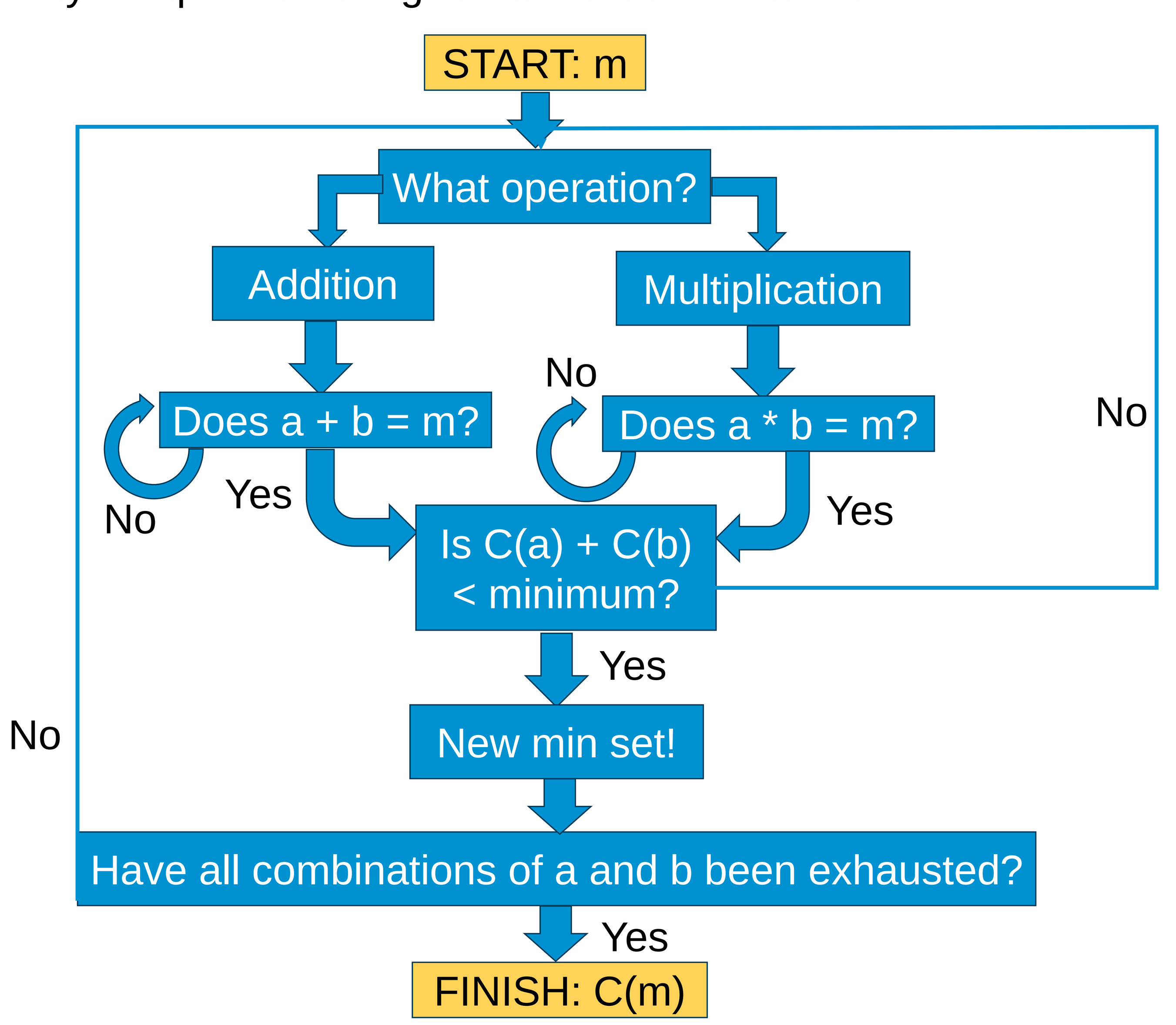

Figure 1. Flowchart of determining the cost of an integer using addition and multiplication.

Cost Function Implementation

Cost Function Framework

- Norfolk's work proposes loose principles for the cost function:
  - Only a finite number of functions can be considered
  - One operation can be evaluated at a time
  - The cost can decrease according to the driver algorithm
- This work adds additional principles for the cost function:
  - Any number of operands can be in an operation (previously 2)
  - The function gives a charge to the operation itself
- **Any iteration (function-wide) has a fixed upper bound!**

Driver Algorithm

- The cost function contains several sub-algorithms
- C(m) can lower by passing m through a sub-algorithm and finding a lower result

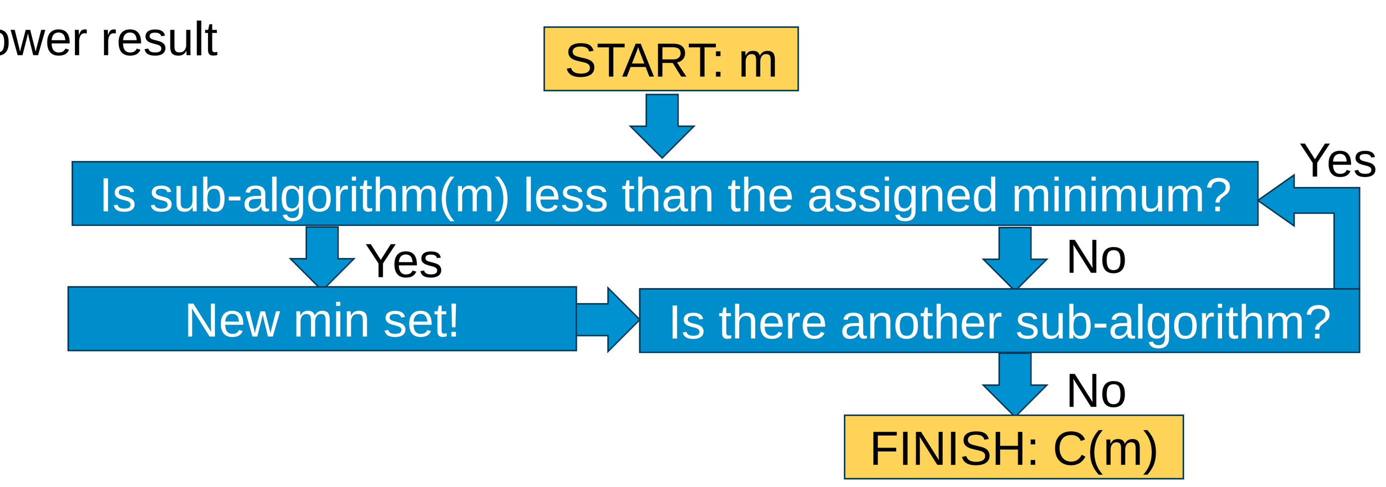

Figure 2. Flow chart explaining the driver algorithm for cost function.

- Sub-algorithms can be used in any order leading to the same result

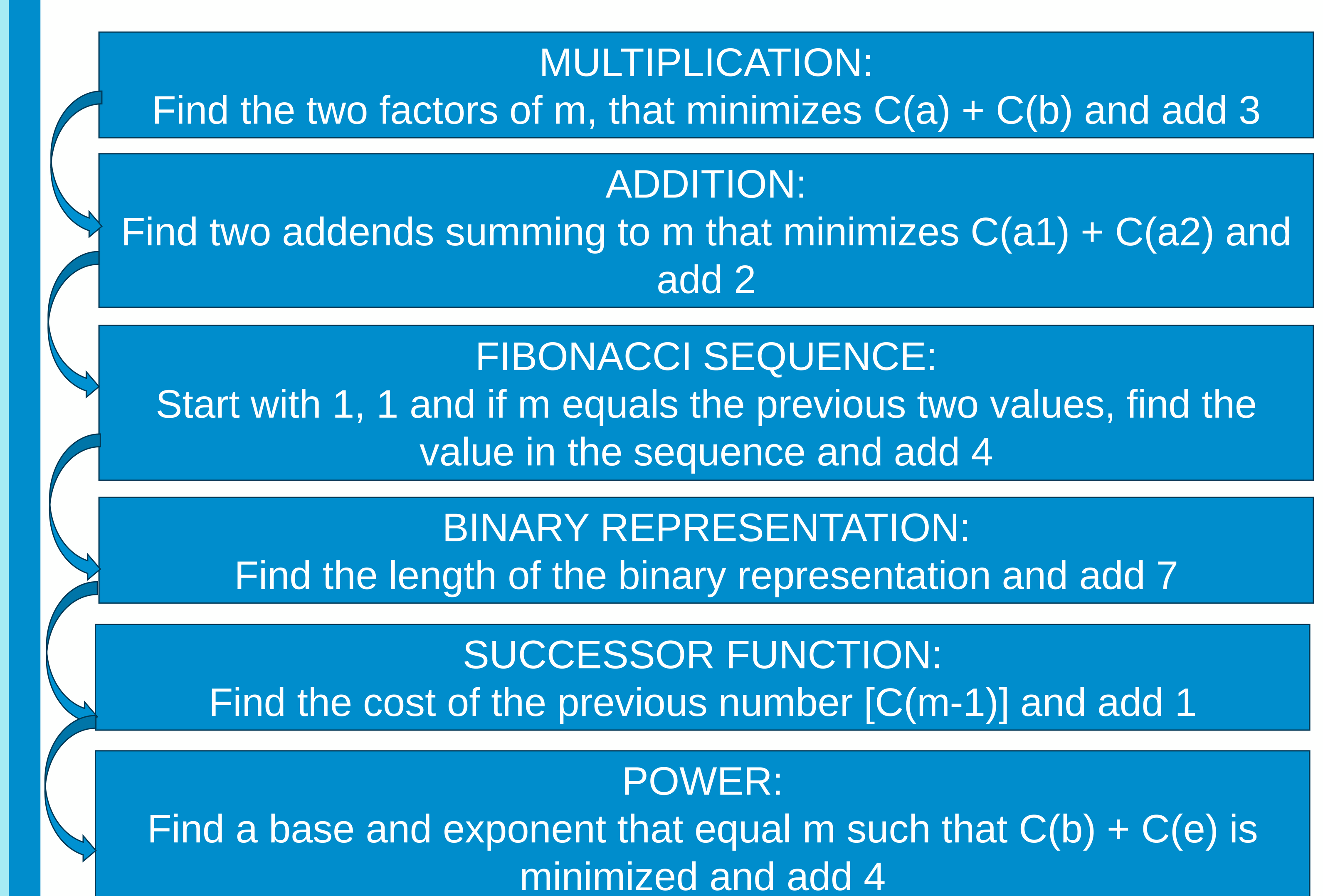

Figure 3. Diagram outlining the cost procedures for each sub-algorithm

Results

- Upper bound of cost function defined by: **$C(m) = \log_2(m) + 7$**
- Both cost function and its upper bound start increasing quickly and increase at a slower rate for higher values of m

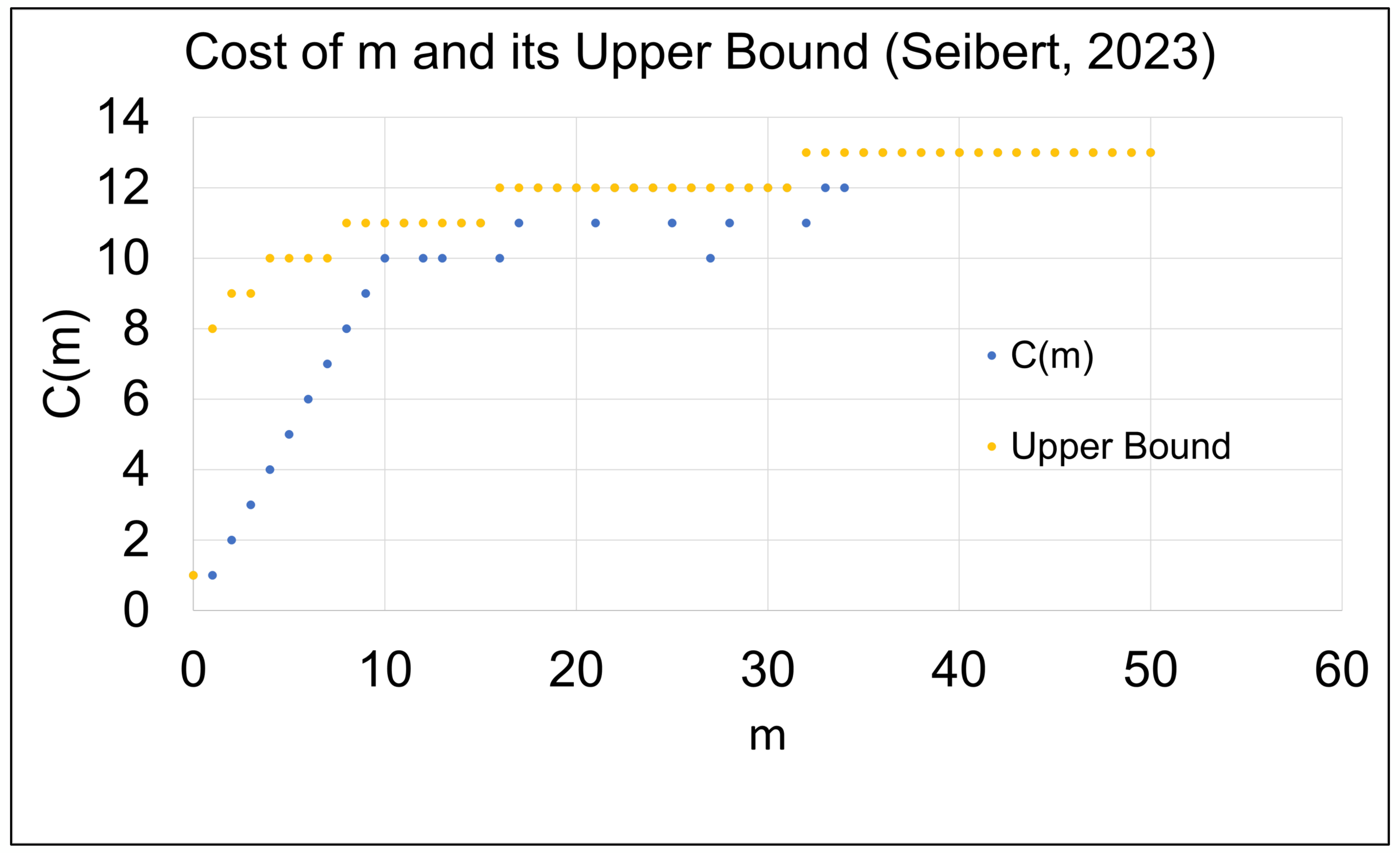

Figure 4. Associated graphs of C(m) and its upper bound against m for non-negative m ≤ 50

Conclusions

- The upper bound is determined by the result of C(m) using solely the binary function
- Subtraction cannot be handled in this program as it relies on cost values higher than numbers passed into the sub-algorithms
- Other setups for the program will need consideration to accommodate function such as subtraction

Reference

- Norfolk, Maxwell (2021) "The Cost of a Positive Integer," Rose-Hulman Undergraduate Mathematics Journal: Vol. 22: Iss. 1, Article 9.

Source Code