

The Sieve of Eratosthenes

In this article, we will learn how to simulate the Sieve of Eratosthenes to find prime numbers.

Concepts used in this program:

- Algorithms
- Arithmetic operators (floor, mod, ceiling, sqrt)
- Conditional looping (repeat until)
- Conditions: YES/NO questions
- Costume-based animation
- Lists
- Logic operators (AND, OR, NOT)
- Nested looping
- Procedures with inputs
- Relational operators ($=$, $<$, $>$)
- Simple looping (repeat, forever)
- String operations (join, letter, length of)
- String variables
- Synchronization using broadcasting
- User input (ASK)
- Variables - local/global scope
- Variables – numbers

Program Description

This program is basically an animation program that shows how the sieve of Eratosthenes works. There is no user interactivity required once the program is started. The animation initially shows a table of numbers 1 to N and then using the technique of Eratosthenes starts dropping numbers until only prime numbers are left in the table.

High-level algorithm:

1. Make a list L of numbers 2 to N.
2. Take the first non-zero number in L as p. If (square of p) > N, stop.
3. Starting from p, replace all multiples of p (except p itself) in L by 0.
4. Go to step 2.

Version 1:

We will first implement this algorithm for $N=64$, and without any animation.

Design:

The above algorithm can be seen as 3 components: first component just doing step 1, the second one doing only step 3, and the final one putting all steps together.

First component (step 1):

First component is straightforward. It takes N as input and builds a list from 1 to N , and then replaces 1 with 0.

Algorithm BuildArray:

```
Input: integer N
Output: list L
I = 1
L = empty
Repeat N
    Add I to L
    I = I + 1
End-repeat
```

Second component (step 3):

Algorithm for ReplaceMultiples:

```
Input: list L (0, 2 to N), integer Q (location of P)
Output: L with all multiples of P (2xP, 3xP, ...) replaced by 0
Steps:
P = item at location Q in L
# Multiples of P will be at Q+P, Q+2P, Q+3P, etc.
Q = Q + P
Repeat until (Q > size of L)
    item at location Q in L = 0
    Q = Q + P
End-repeat
```

Third component:

The following steps wrap up all 3 steps (2, 3, and 4).

```
Q = 2 (index of first non-zero item in L)
P = item at location Q in L
Repeat until (Q is out of range OR (square of P) > N)
    ReplaceMultiples(Q)
    Q = Q + 1
    Repeat until (Q is out of range OR item at Q in L > 0)
        Q = Q + 1
End-repeat
```

Solution: Eratosthenes-1.sb2

<https://scratch.mit.edu/projects/318675682/>

Version 2:

Let's animate the algorithm using the "Number table" idea (see the "Number-table" program in "Practice CS Concepts with Scratch" to understand how to design it).

Follow these steps to animate the algorithm:

- Take the Number-table program. Modify it work for the range N=5 to 257 (anything bigger would look too tiny).
- Import the code of version 1 above. (You can export the sprites and import them). You will need to call the "Eratosthenes" script after displaying the initial array of numbers. As the algorithm goes on replacing numbers with 0s, you will need to update the display. See below for the "ResetCell" algorithm. Instead of writing 0, we will just use a blank square.

Algorithm ResetCell:

Input: integer cell (1 to N), integer tableSize (table dimension), integer cellSize (width of the square sprite), integer startx and starty (x,y of the first cell)

Steps:

row = 1 + integer division((cell - 1) / tableSize)

column = 1 + remainder((cell - 1) / tableSize)

X = startx + (cellSize * (Column-1))

Y = starty - (cellSize * (Row-1))

Go to X, Y

Stamp blank square

Solution: Eratosthenes-final.sb2

<https://scratch.mit.edu/projects/318675993/>

Author: Abhay B. Joshi (abjoshi@yahoo.com)

Last updated: 2 July 2019