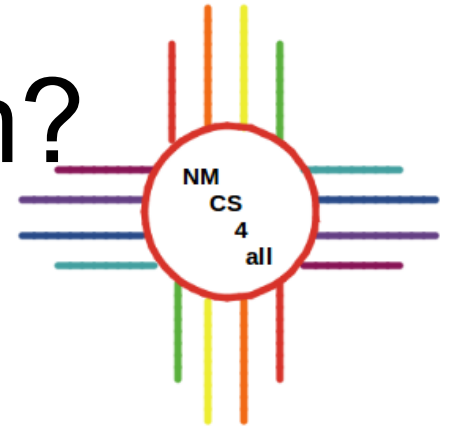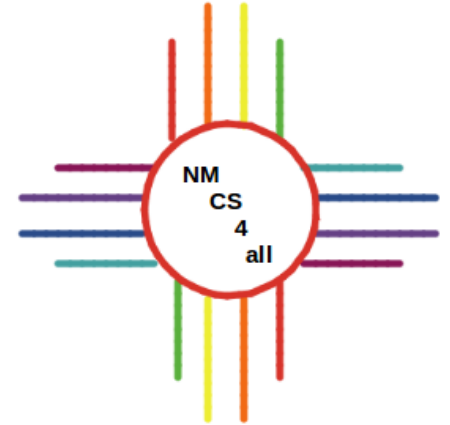# Module 6

Overview

- Algorithms

- Sorting algorithms

- Agentsets and lists

- Looping in NetLogo

What is an algorithm?

- A set of steps to accomplish a task

- Varying complexity

    - Input data

    - Calculations

    - Result
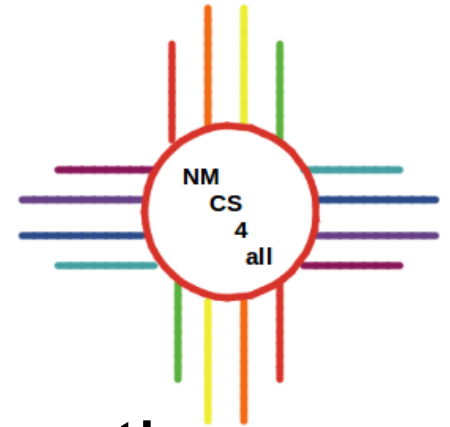
Using Algorithms

- Existing algorithms can be used

- Save yourself some effort

- Make your programs faster

Examples

- Possible outcomes

- Simulations

- Search and analyze

- Make decisions
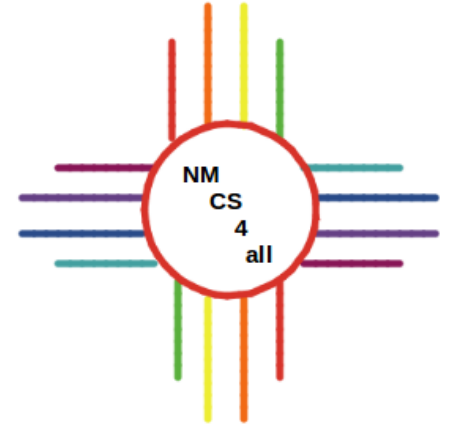
Wiggle Walk

- Turtles always follow the same instructions

```
to wiggleWalk
    ask turtles
    [
        left random 91
        right random 91
        forward 1
    ]
    tick
end
```
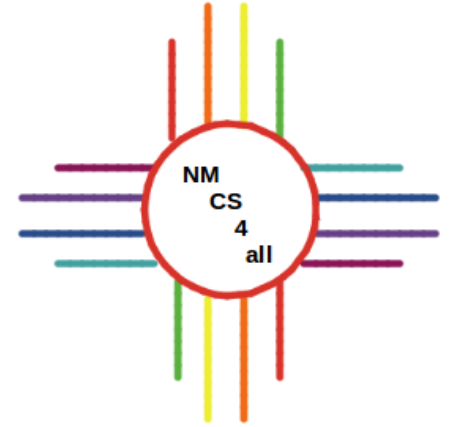
Good Algorithms

- Correctness

    - The result should be correct

- Efficiency

    - We will accept results that might not be the

    best for the sake of efficiency

Categories

- Counting

- Sorting

- Searching

- Mapping and Graphing

- Encryption

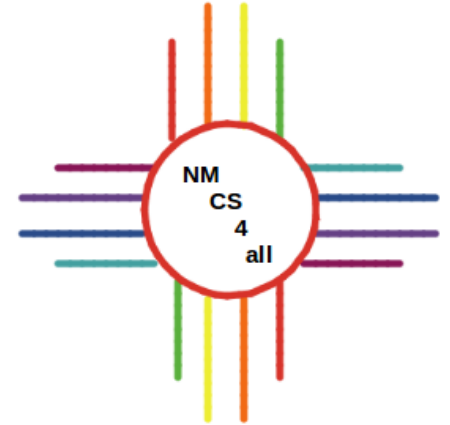- Packing

- Maze

Big O

- Used in computer science

- Analyzing algorithms

- Measuring how they run as the input size is

increased dramatically

    - Processing time

    - Memory used
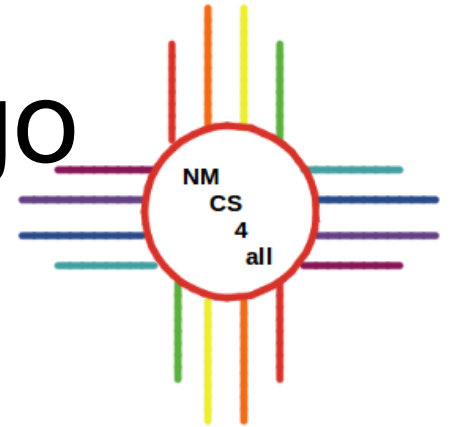
Sorting Algorithms

- An important type of algorithm

- Arrange things in order based on some criteria

- Examples

    - Insertion Sort

    - Merge Sort

Lists and Sets

- Data: scalars and compositions

    - Lists

        - Ordered collection of values

    - Sets

        - Unordered collection with no

repetitions

Agentsets in NetLogo

- Sets with agents of a common general type

    - Turtles, patches and links
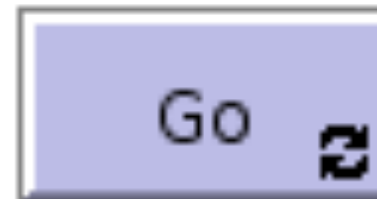
    - Breeds

    - Variables

Loops in NetLogo

- Infinite loops

- Counted loops

- Conditional loops

Infinite Loops

- Repeat until:

    - A stop command

    - Manually turned off

    - A runtime error

Go

Counted Loops

- Repeat a fixed number of times

- Must know how many times

    - Before programming

    - Be able to calculate

```
to counted
  create-turtles 1
  ask turtles
  [
    repeat 30
    [
      left random 91
      right random 91
      forward 1
    ]
  ]
end
```

Conditional Loops

- Repeat while a conditional statement is true

- Must be true to start

- Must become false

```
to conditional
  create-turtles 1
  ask turtles
  [
    while [random 85 < 80]
    [
      left random 91
      right random 91
      forward 1
    ]
  ]
end
```

Thank you for watching!

Video created by Bianca Bologa

https://moseslab.cs.unm.edu/lab-page/bianca-bologa.html