An Introduction to Data Structures

Measuring Efficiency with BigO Notation

Measuring Efficiency with BigO Notation - Introduction

- We want a quantifiable way to measure how efficient certain data structures are at different tasks we might ask of it
  - Searching through
  - Modifying
  - Accessing

Measuring Efficiency with BigO Notation - Introduction

- We want a quantifiable way to measure how efficient certain data structures are at different tasks we might ask of it
    - Searching through
    - Modifying
    - Accessing

Measuring Efficiency with BigO Notation - Introduction

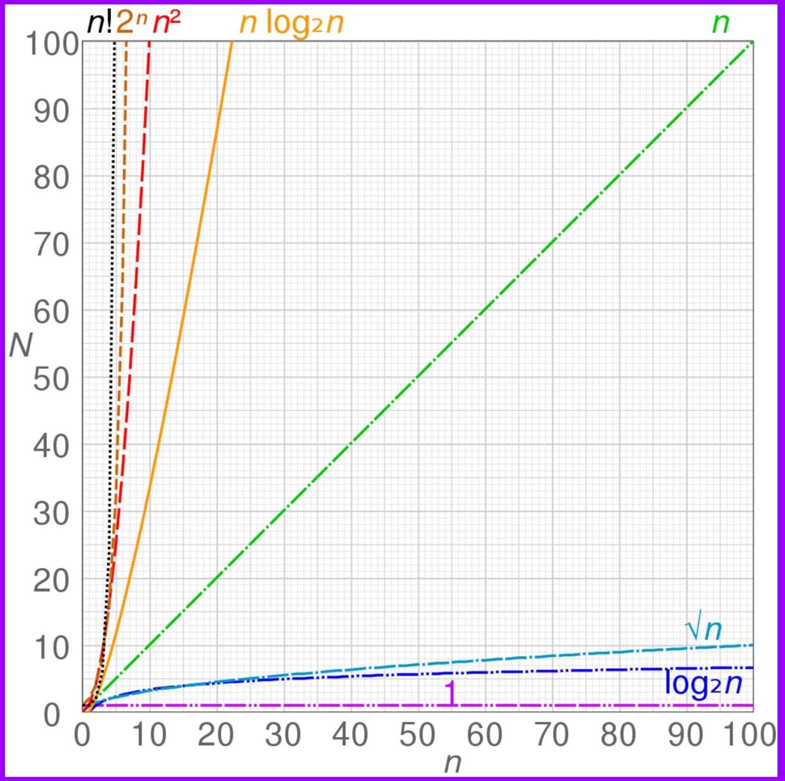
- We want a quantifiable way to measure how efficient certain data structures are at different tasks we might ask of it
    - Searching through
    - Modifying
    - Accessing

Measuring Efficiency with BigO Notation - Introduction

- We want a quantifiable way to measure how efficient certain data structures are at different tasks we might ask of it
  - Searching through
  - Modifying
  - Accessing

Measuring Efficiency with BigO Notation - Introduction

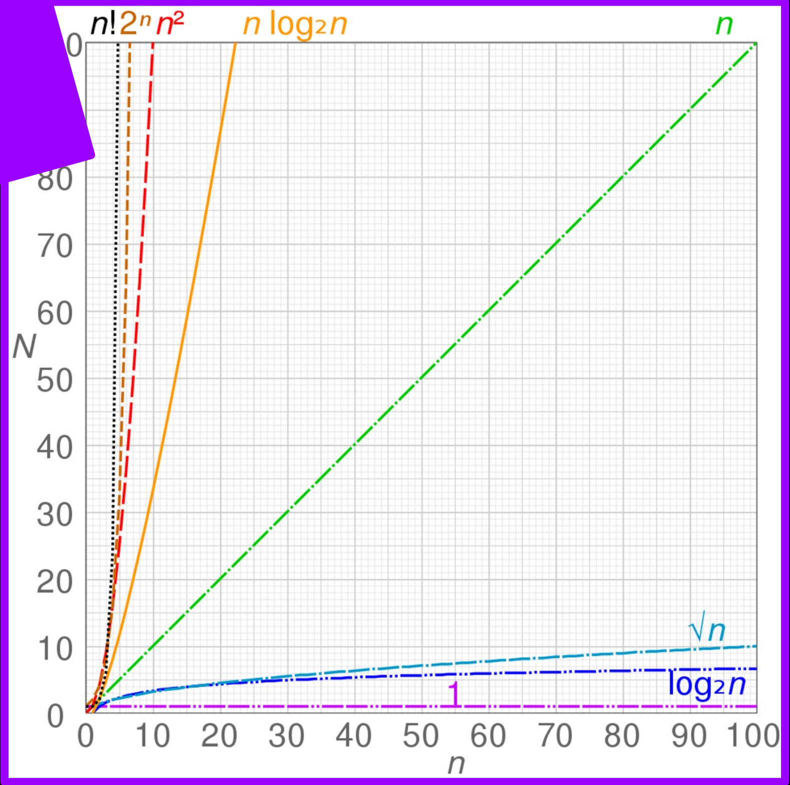
- The **industry standard** for this kind of measurement
  - BigO Notation



Measuring Efficiency with BigO Notation - Introduction

WHAT IS IT?

- y **standard** for is kind of measurement
    - ○ BigO Notation

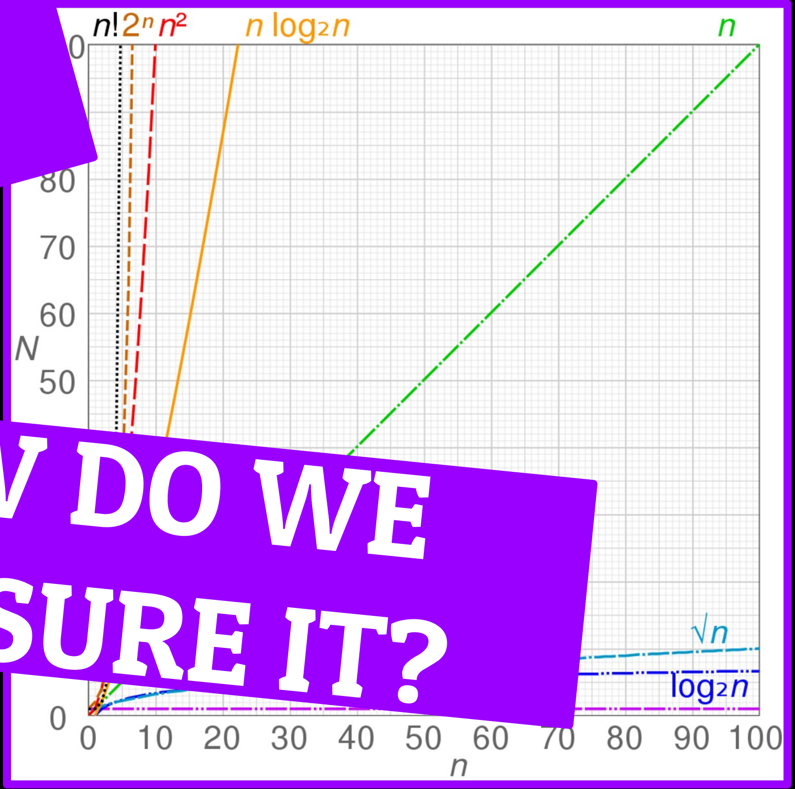


Measuring Efficiency with BigO Notation - Introduction

WHAT IS IT?

- y standard for his kind of measurement
    - BigO Notation

HOW DO WE MEASURE IT?

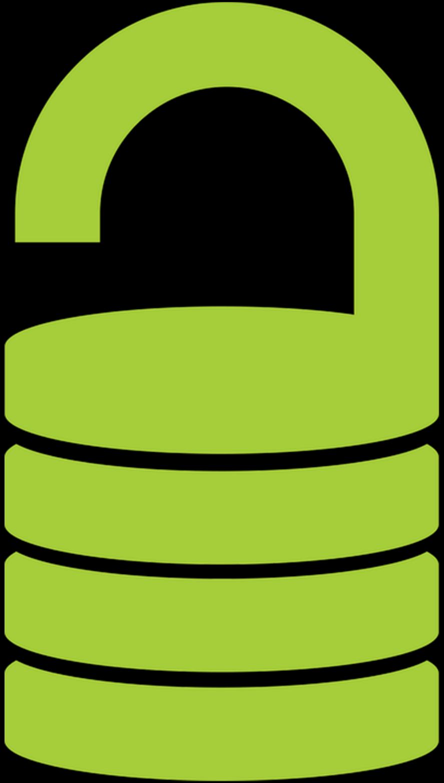


Measuring Efficiency with BigO Notation - Introduction

- **BigO Notation**
  - A way to basically "**Score**" a data structure based on **4 criteria**
  - The most **common functions** you might want from a data structure
    - Accessing elements
    - Searching for an element
    - Inserting an element
    - Deleting an element

Measuring Efficiency with BigO Notation - Introduction

- **BigO Notation**
  - A way to basically "**Score**" a data structure based on **4 criteria**
  - The most **common functions** you might want from a data structure
    - **Accessing elements**
    - Searching for an element
    - Inserting an element
    - Deleting an element

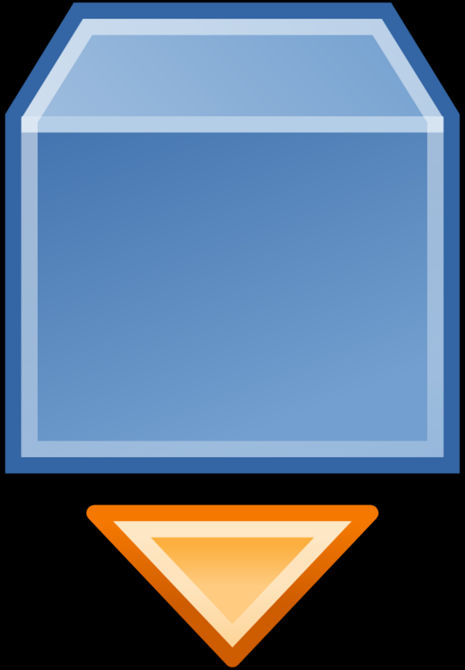


Measuring Efficiency with BigO Notation - Introduction

- BigO Notation
  - A way to basically "Score" a data structure based on 4 criteria
  - The most common functions you might want from a data structure
    - Accessing elements
    - Searching for an element
    - Inserting an element
    - Deleting an element

Measuring Efficiency with BigO Notation - Introduction

- **BigO Notation**
    - A way to basically "**Score**" a data structure based on **4 criteria**
    - The most **common functions** you might want from a data structure
        - Accessing elements
        - Searching for an element
        - **Inserting an element**
        - Deleting an element



Measuring Efficiency with BigO Notation - Introduction

- **BigO Notation**
  - A way to basically “**Score**” a data structure based on **4 criteria**
  - The most **common functions** you might want from a data structure
    - Accessing elements
    - Searching for an element
    - Inserting an element
    - **Deleting an element**

Measuring Efficiency with BigO Notation - Introduction

- By measuring how efficiently a data structure can do these 4 things
    - We can create a “report card”

Measuring Efficiency with BigO Notation - Introduction

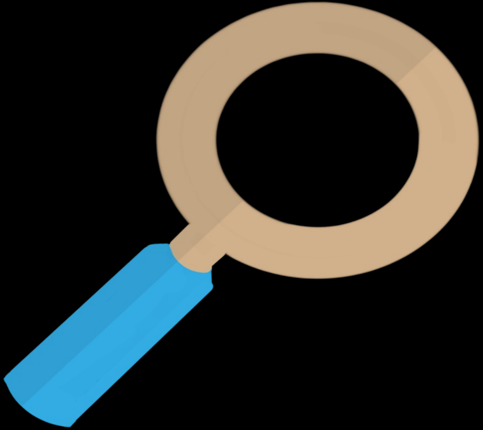
- By measuring how **efficiently** a data structure can do these 4 things
    - We can create a “**report card**”

Accessing

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - ○ We can create a "**report card**"

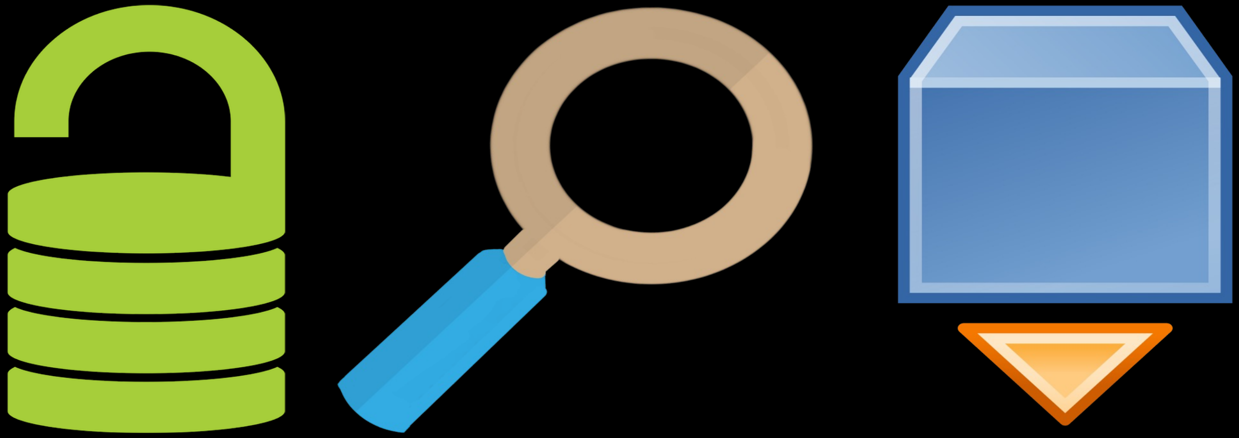
Accessing



Searching

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a "**report card**"



Accessing

Searching

Inserting

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a “**report card**”

Accessing

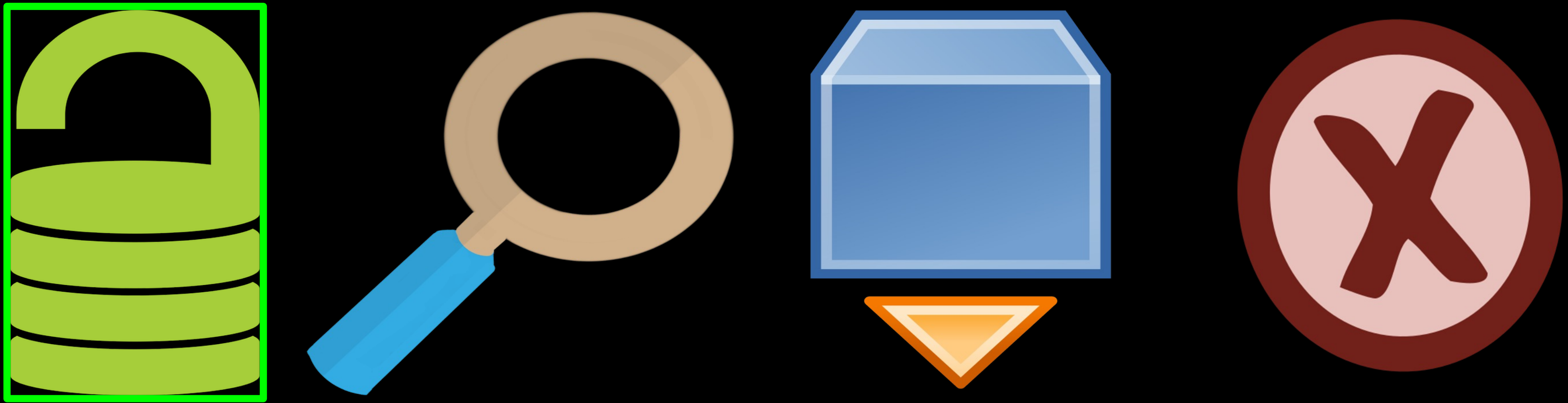
Searching

Inserting

Deleting

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a "**report card**"



Accessing

Searching

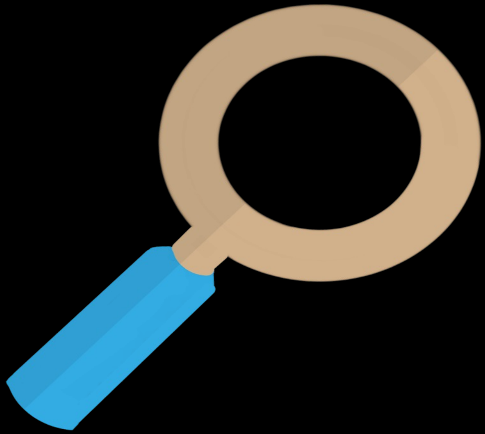
Inserting

Deleting

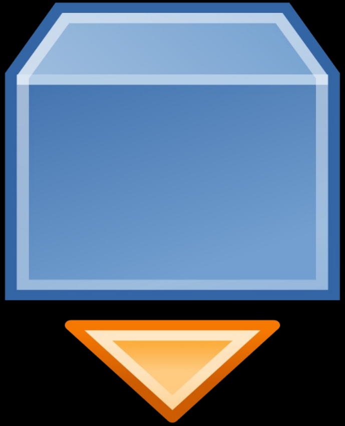
Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a "**report card**"

Accessing



Searching

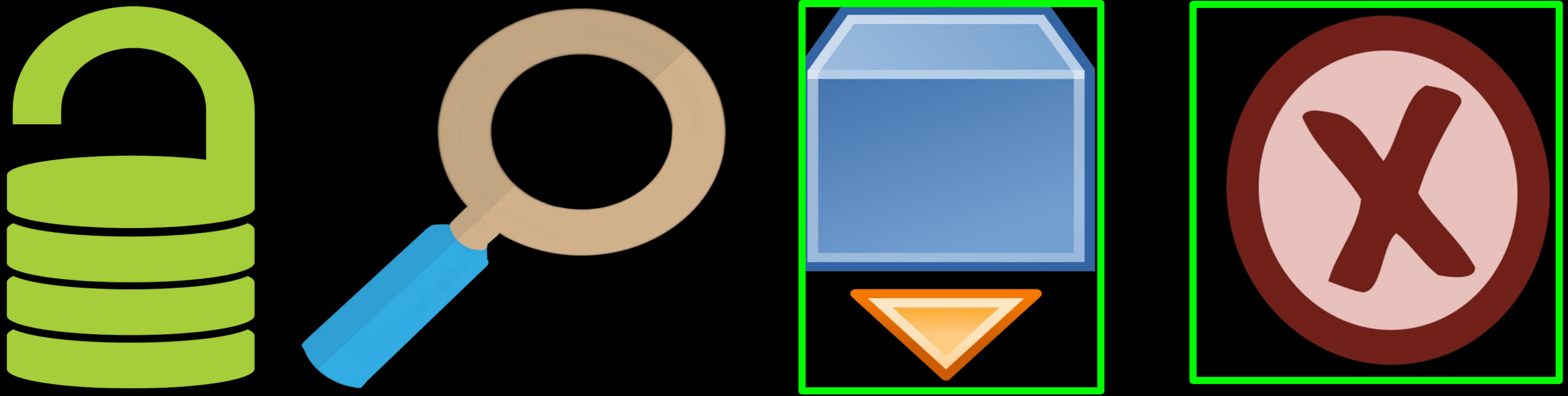


Inserting

Deleting

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a “**report card**”



Accessing

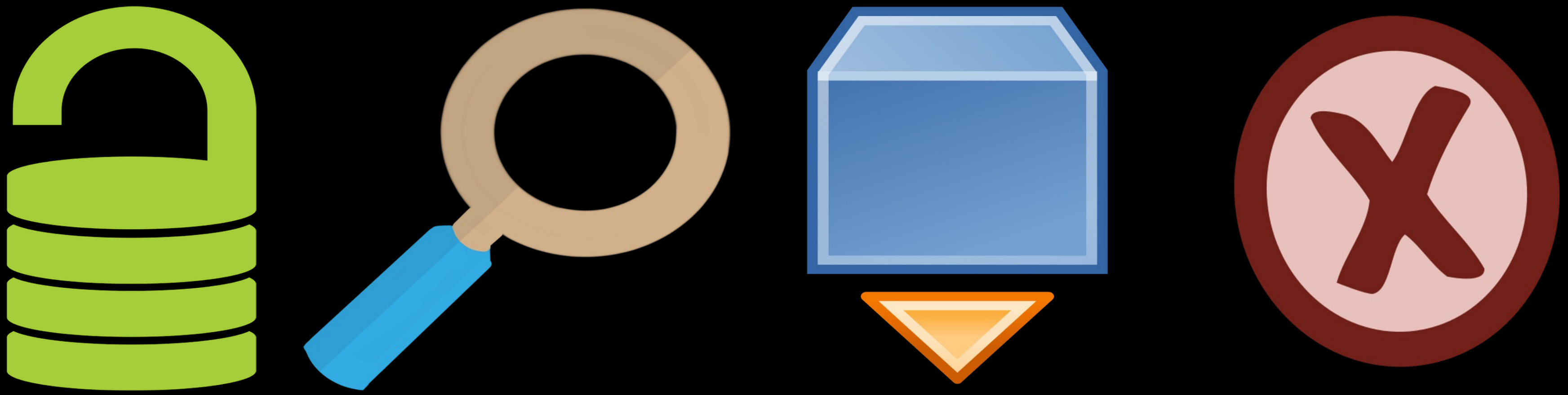
Searching

Inserting

Deleting

Measuring Efficiency with BigO Notation - Introduction

- By measuring how **efficiently** a data structure can do these 4 things
  - We can create a “**report card**”



Accessing **Searching** **Inserting** **Deleting**

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A Time Complexity Equation works by inserting the size of the data-set as an integer n, and returning the number of operations that need to be conducted by the computer before the function can finish

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A Time Complexity Equation works by inserting the size of the data-set as an integer n, and returning the number of operations that need to be conducted by the computer before the function can finish

N = The Size of the Data Set (Amount of elements contained within the Data Structure)

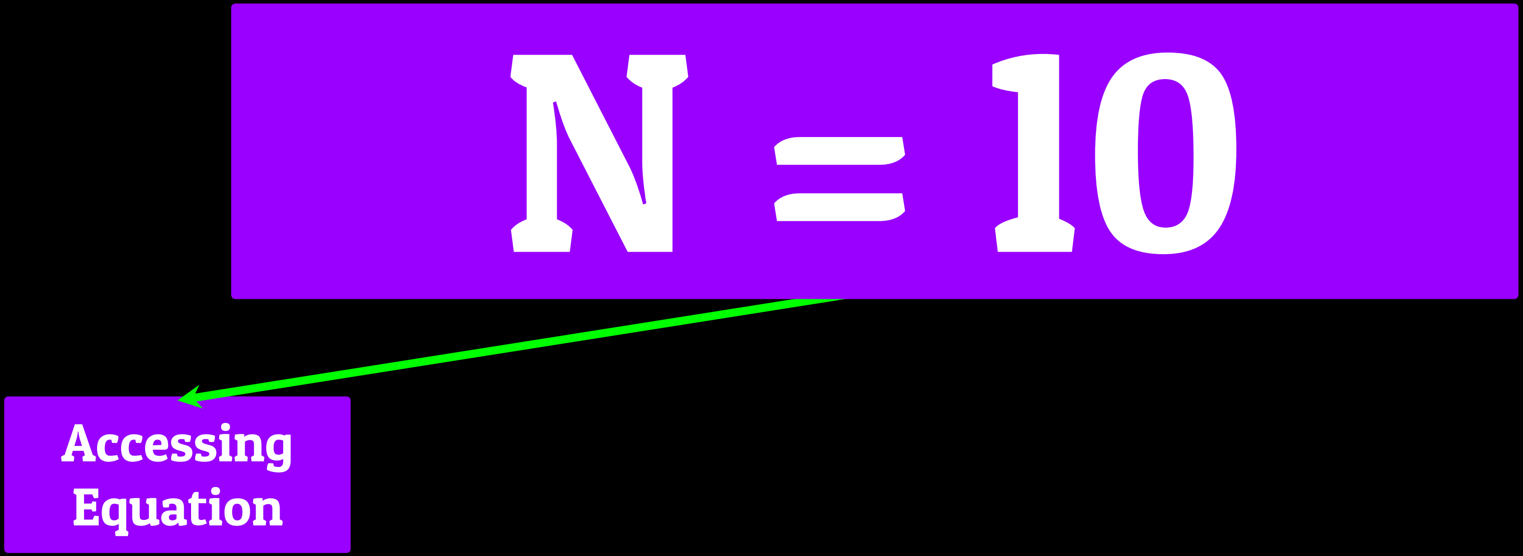
Measuring Efficiency with BigO Notation - Time Complexity Equations

- A **Time Complexity Equation** works by inserting the **size** of the data-set as an integer **n**, and returning the number of **operations** that need to be conducted by the computer before the function can finish

N = 10

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A **Time Complexity Equation** works by inserting the **size** of the data-set as an integer **n**, and returning the number of **operations** that need to be conducted by the computer before the function can finish

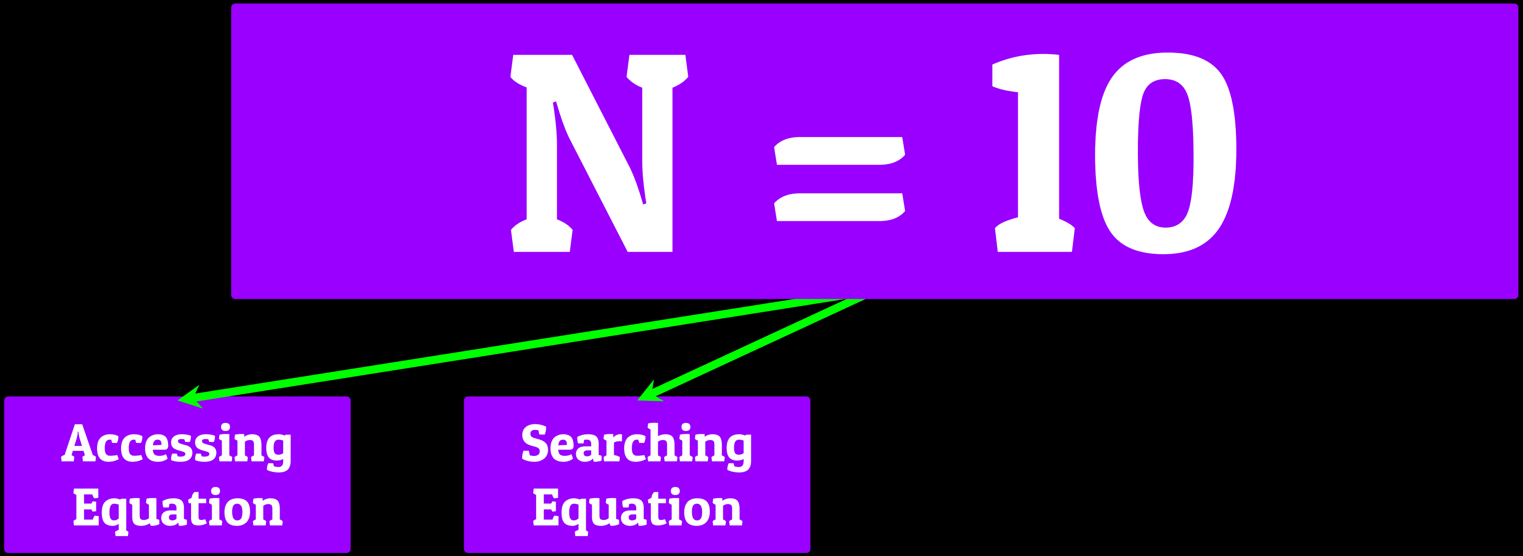


N = 10

Accessing Equation

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A Time Complexity Equation works by inserting the size of the data-set as an integer n, and returning the number of operations that need to be conducted by the computer before the function can finish



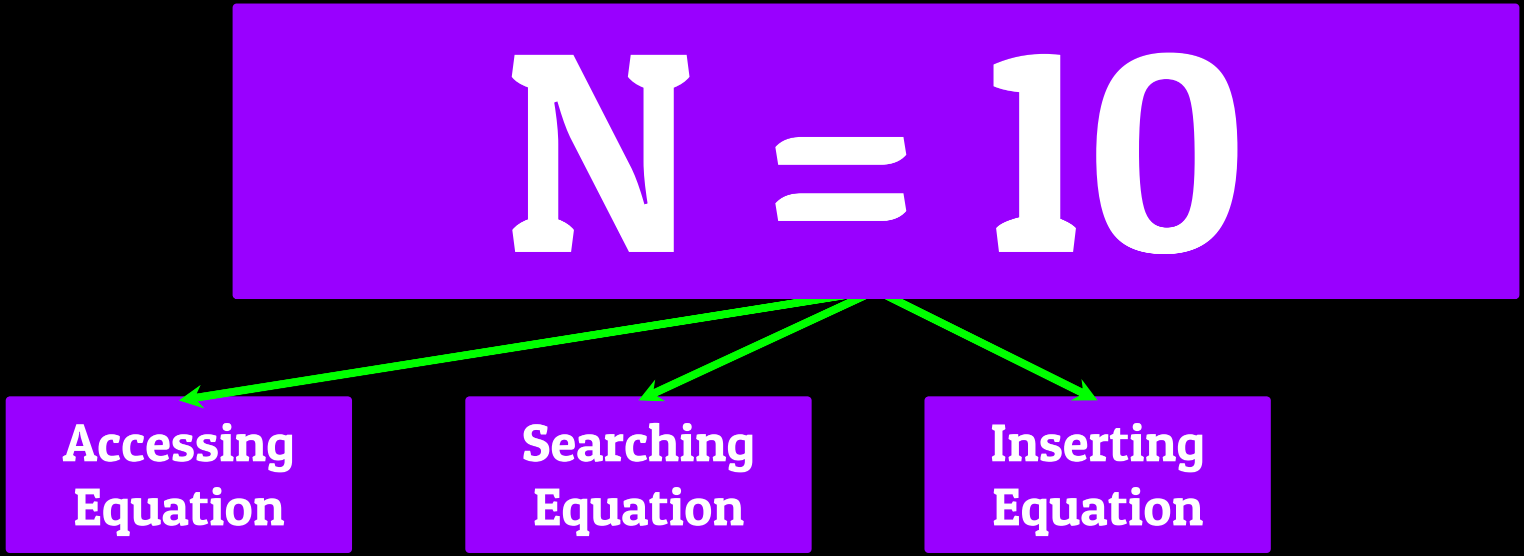
N = 10

Accessing Equation

Searching Equation

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A **Time Complexity Equation** works by inserting the **size** of the data-set as an integer **n**, and returning the number of **operations** that need to be conducted by the computer before the function can finish



N = 10

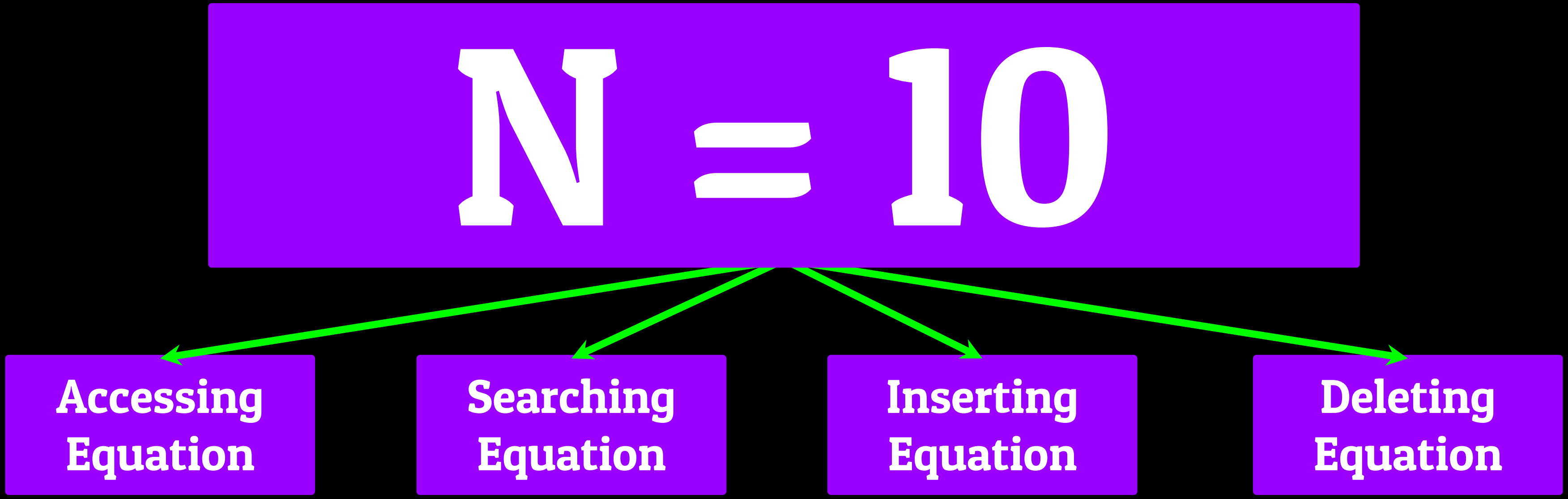
Accessing Equation

Searching Equation

Inserting Equation

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A **Time Complexity Equation** works by inserting the **size** of the data-set as an integer **n**, and returning the number of **operations** that need to be conducted by the computer before the function can finish



N = 10

Accessing Equation

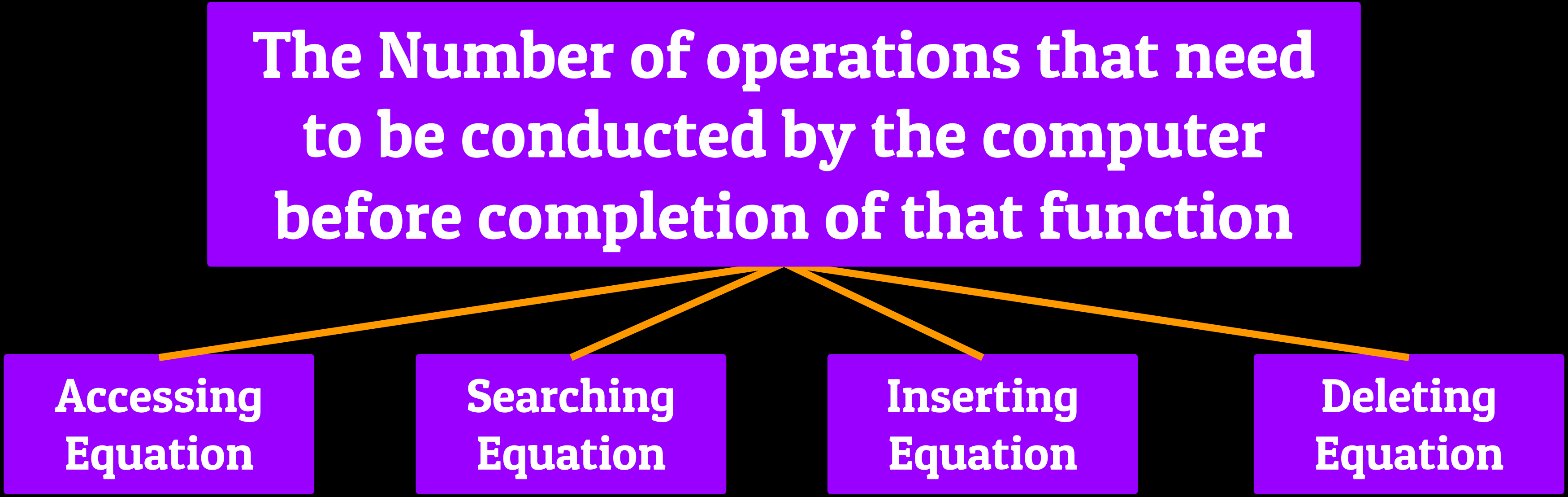
Searching Equation

Inserting Equation

Deleting Equation

Measuring Efficiency with BigO Notation - Time Complexity Equations

- A Time Complexity Equation works by inserting the size of the data-set as an integer n, and returning the number of operations that need to be conducted by the computer before the function can finish



The Number of operations that need to be conducted by the computer before completion of that function

Accessing Equation

Searching Equation

Inserting Equation

Deleting Equation

Measuring Efficiency with BigO Notation - Time Complexity Equations

- We always use the **worst-case scenario** when judging these data structures

8 Operations

2 Operations

50 Operations

42 Operations

1898 Operations

Measuring Efficiency with BigO Notation - Time Complexity Equations

- We always use the **worst-case scenario** when judging these data structures

8 Operations

2 Operations

50 Operations

42 Operations

1898 Operations

Measuring Efficiency with BigO Notation - Time Complexity Equations

- We always use the **worst-case scenario** when judging these data structures

8 Operations

2 Operations

50 Operations

42 Operations

1898 Operations

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It's called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function

Measuring Efficiency with BigO Notation - The Meaning of BigO

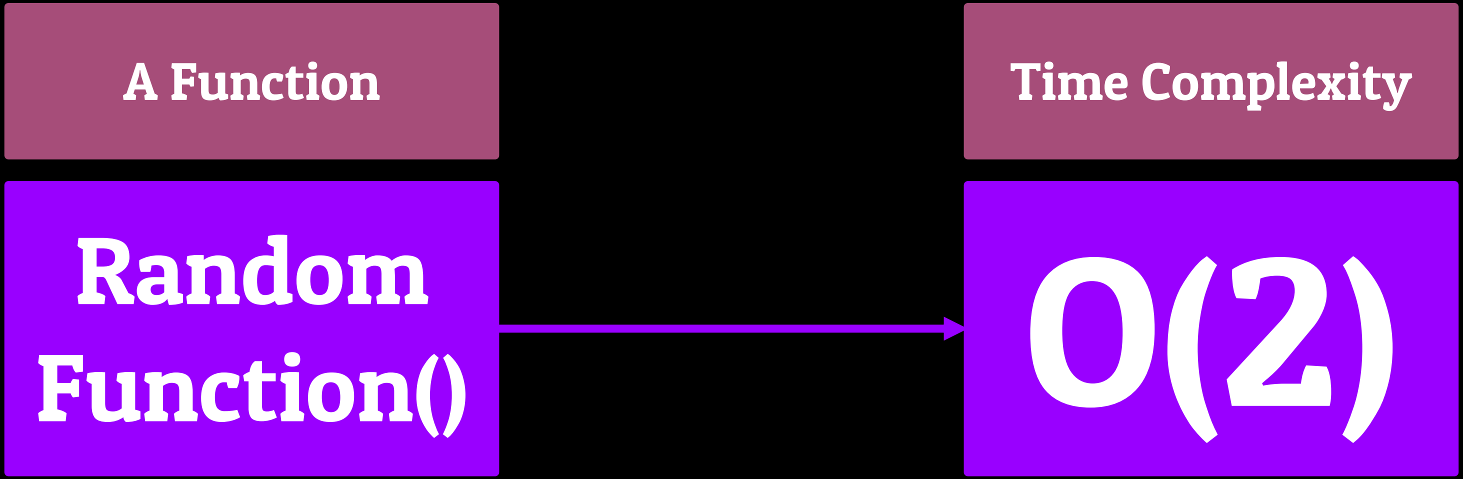
- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function

A Function

Random Function()

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function



A Function

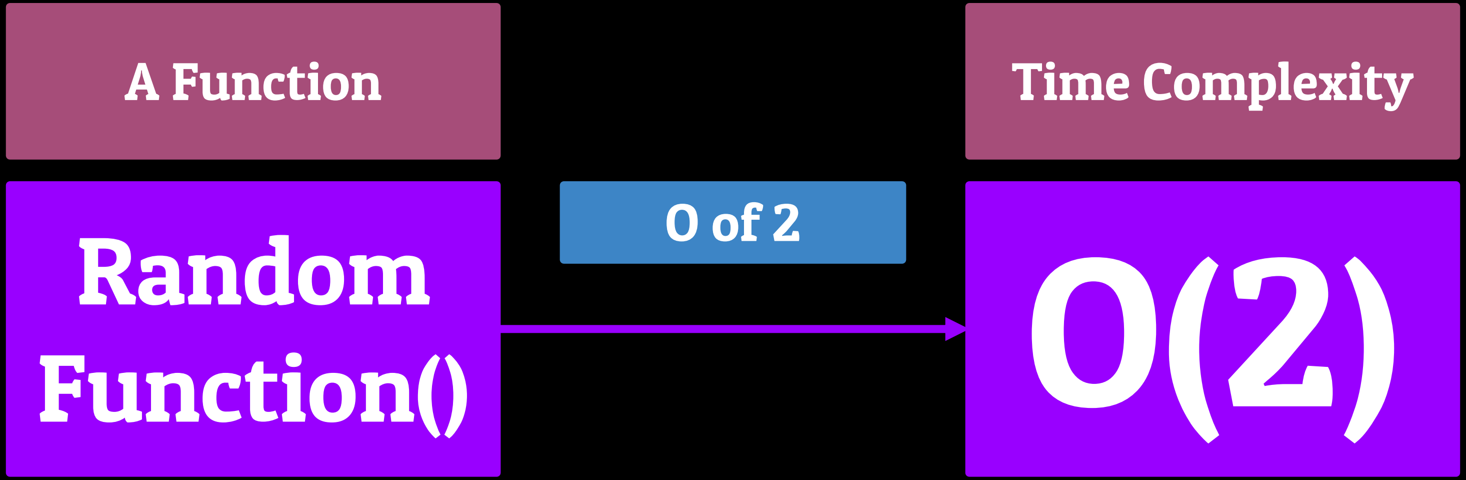
Random Function()

Time Complexity

O(2)

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function



A Function

Time Complexity

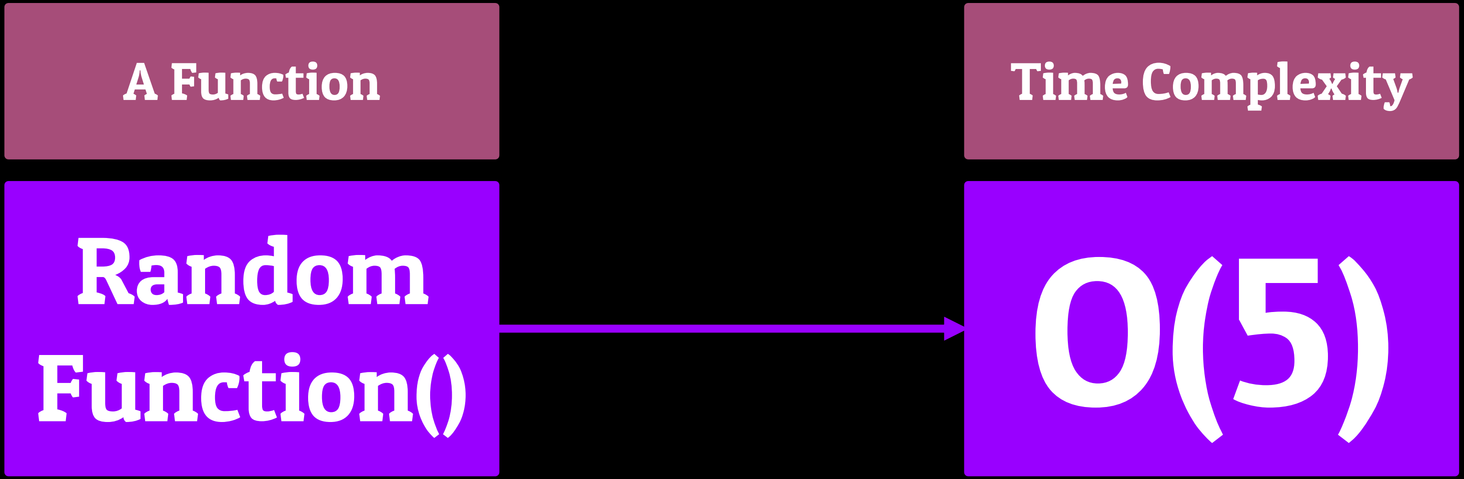
Random Function()

O of 2

O(2)

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function



A Function

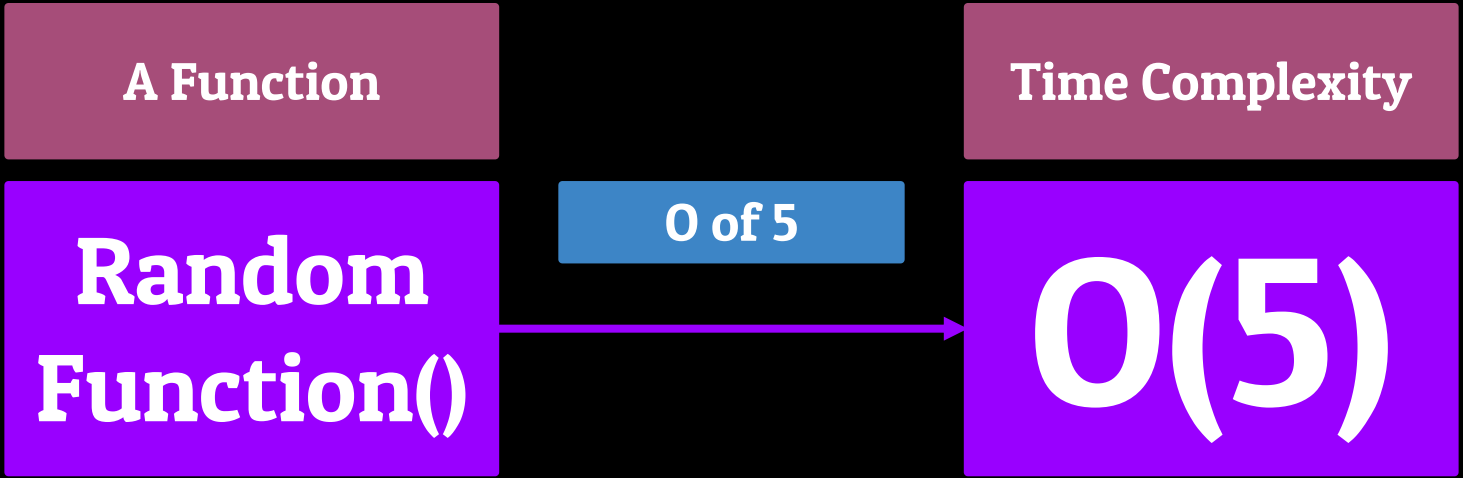
Random Function()

Time Complexity

O(5)

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function



A Function

Time Complexity

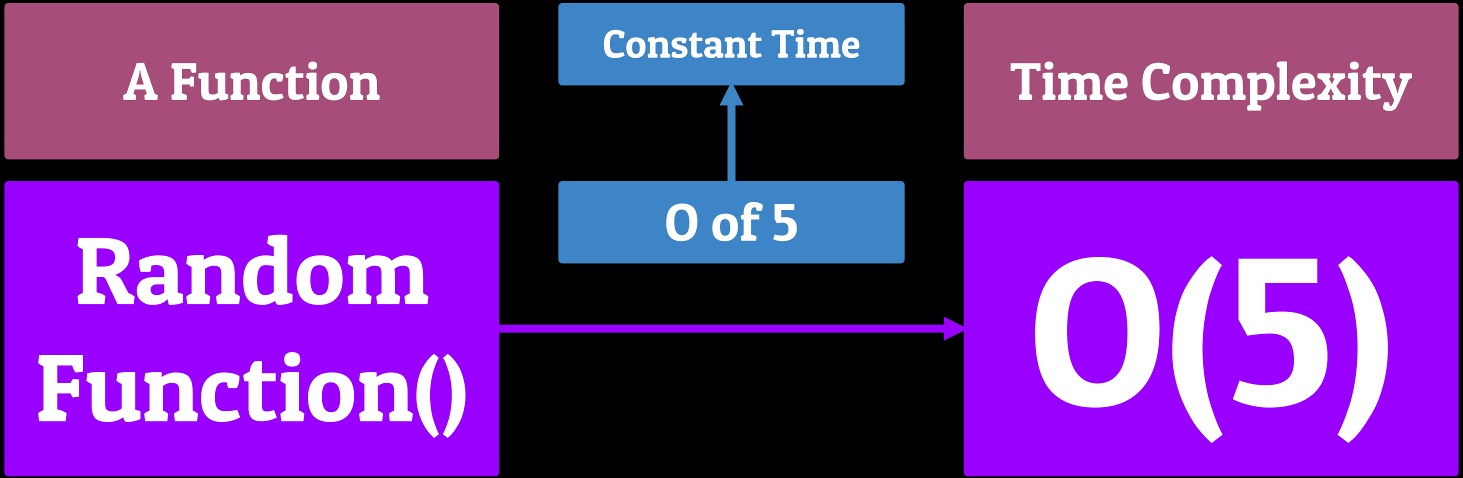
Random Function()

O of 5

O(5)

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
  - The parenthesis houses the function



A Function

Constant Time

Time Complexity

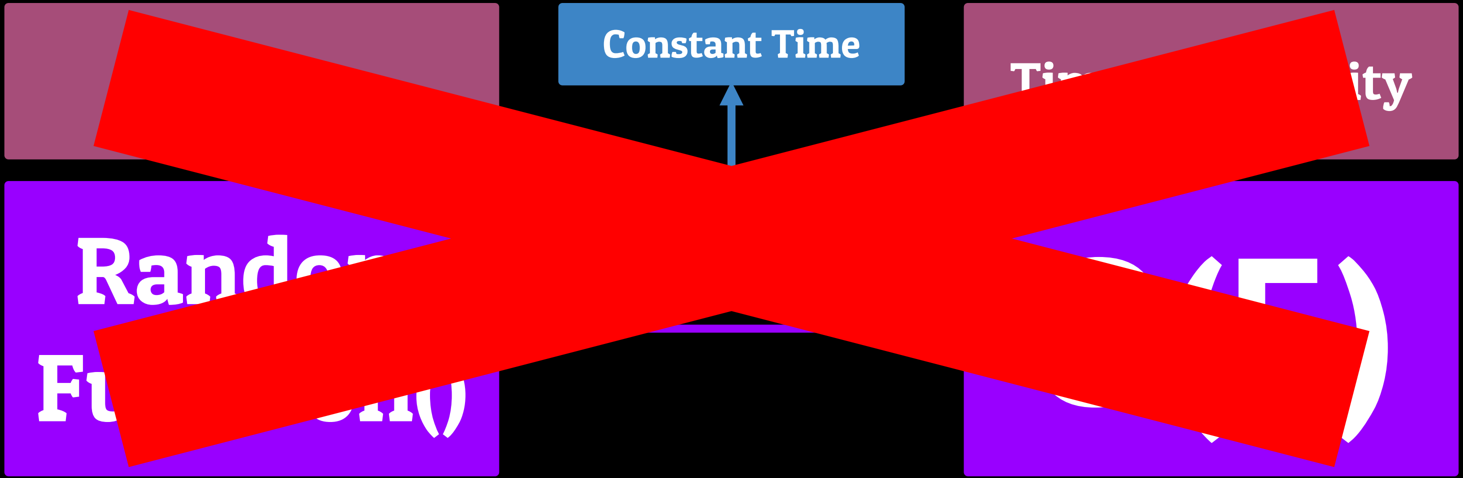
Random Function()

O of 5

O(5)

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
  - The parenthesis houses the function



Measuring Efficiency with BigO Notation - The Meaning of BigO

- It's called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
  - The parenthesis houses the function

Most of the time, our integer n, is going to have some adverse-effect on how many operations it takes

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
    - The parenthesis houses the function

Most of the time, our integer n, is going to have some adverse-effect on how many operations it takes

Larger Data Set

Measuring Efficiency with BigO Notation - The Meaning of BigO

- It’s called BigO notation because the syntax for the Time Complexity equations includes a BigO and then a set of parentheses
  - The parenthesis houses the function

Most of the time, our integer n, is going to have some adverse-effect on how many operations it takes

Larger Data Set

More Instructions

Measuring Efficiency with BigO Notation - Why BigO?

- We measure efficiency in # of operations performed because measuring by how long the function takes to run would be silly
    - Measuring by time is biased towards better hardware

Measuring Efficiency with BigO Notation - Why BigO?

- We measure efficiency in # of operations performed because measuring by how long the function takes to run would be silly
  - Measuring by time is biased towards better hardware

Measuring Efficiency with BigO Notation - Why BigO?

- We measure efficiency in # of operations performed because measuring by how long the function takes to run would be silly
  - Measuring by time is biased towards better hardware

Measuring Efficiency with BigO Notation - Why BigO?

- We measure efficiency in # of operations performed because measuring by how long the function takes to run would be silly
  - Measuring by time is biased towards better hardware

›

Speed

Measuring Efficiency with BigO Notation - Why BigO?

- We measure efficiency in # of operations performed because measuring by how long the function takes to run would be silly
  - Measuring by time is biased towards better hardware

=

Operations

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics

Accessing

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics

Accessing

Searching

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics

Accessing

Searching

Inserting

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics

Accessing

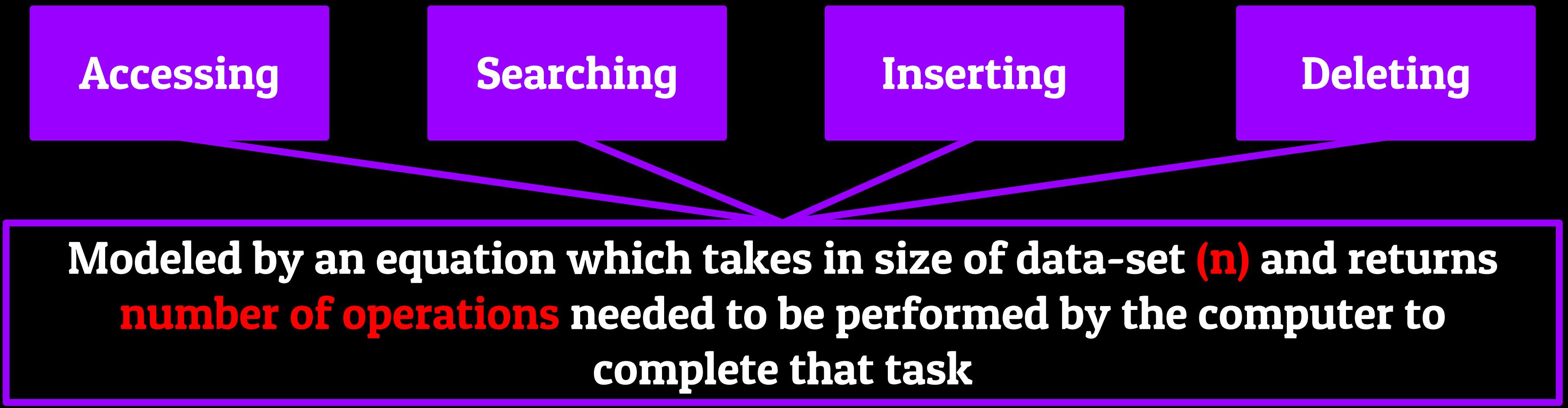
Searching

Inserting

Deleting

Measuring Efficiency with BigO Notation - Quick Recap

We measure efficiency based on 4 metrics



Accessing

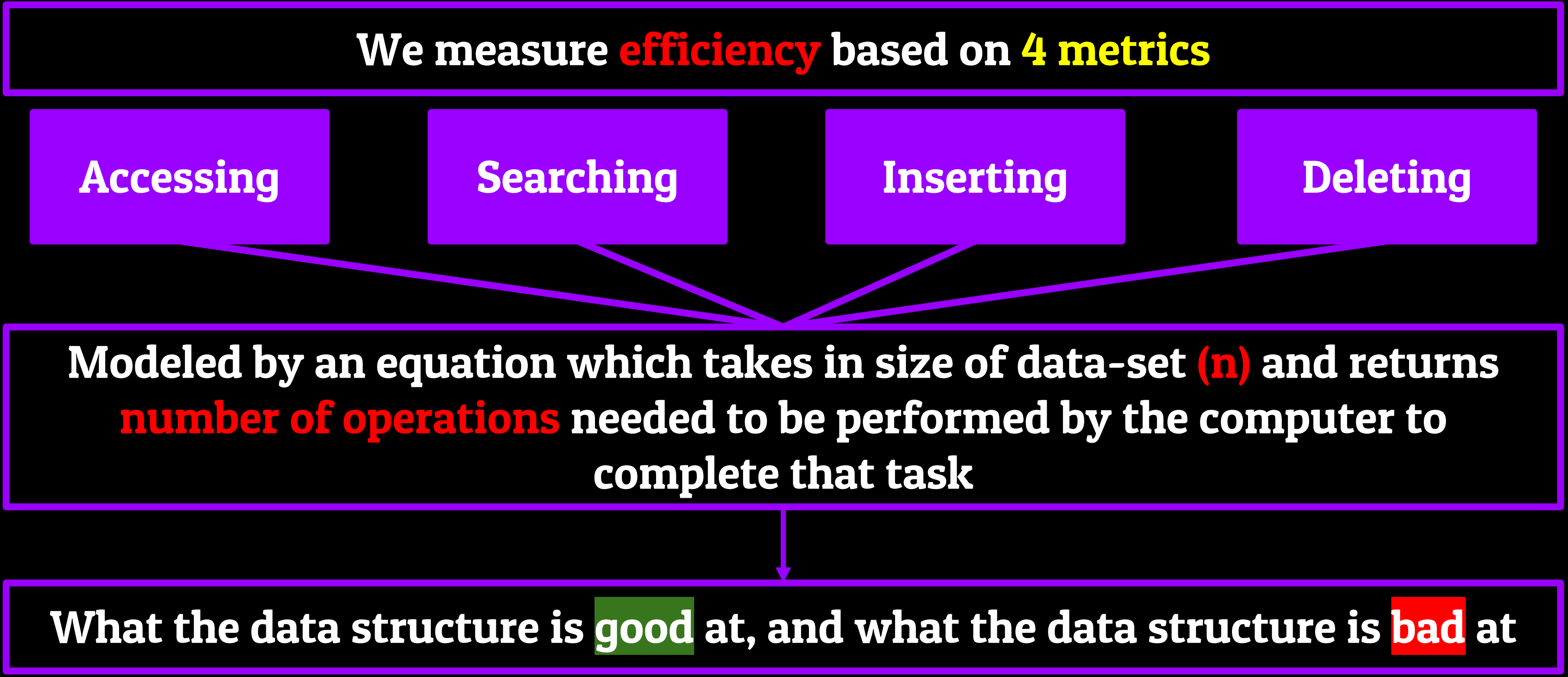
Searching

Inserting

Deleting

Modeled by an equation which takes in size of data-set (n) and returns number of operations needed to be performed by the computer to complete that task

Measuring Efficiency with BigO Notation - Quick Recap



We measure efficiency based on 4 metrics

Accessing

Searching

Inserting

Deleting

Modeled by an equation which takes in size of data-set (n) and returns number of operations needed to be performed by the computer to complete that task

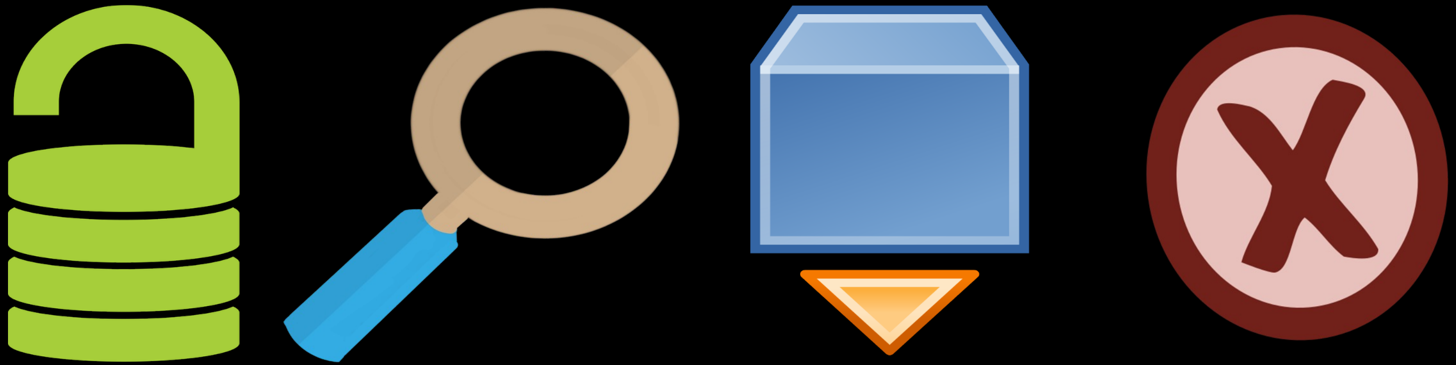
What the data structure is good at, and what the data structure is bad at

Measuring Efficiency with BigO Notation - Quick Recap

- A Data Structures efficiency isn’t the end all be all for deciding on which data structure to use in your program
  - Some have specific quarks or features which separate them

Measuring Efficiency with BigO Notation - Quick Recap

- A Data Structures **efficiency** isn't the end all be all for deciding on which data structure to use in your program
  - Some have specific **quarks** or **features** which separate them



Accessing

Searching

Inserting

Deleting

Measuring Efficiency with BigO Notation - Quick Recap

- A Data Structures efficiency isn’t the end all be all for deciding on which data structure to use in your program
  - Some have specific quarks or features which separate them

Should not be the ONLY metric used

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations mean in terms of efficiency
  - 6 most common Time Complexity Equations

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations mean in terms of efficiency
    - 6 most common Time Complexity Equations

O(1)

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations **mean** in terms of **efficiency**
    - 6 **most common** Time Complexity Equations

O(1)

O(n)

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations **mean** in terms of **efficiency**
  - 6 **most common** Time Complexity Equations

O(1)

O(n)

O(log n)

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations **mean** in terms of **efficiency**
  - 6 **most common** Time Complexity Equations

O(1)

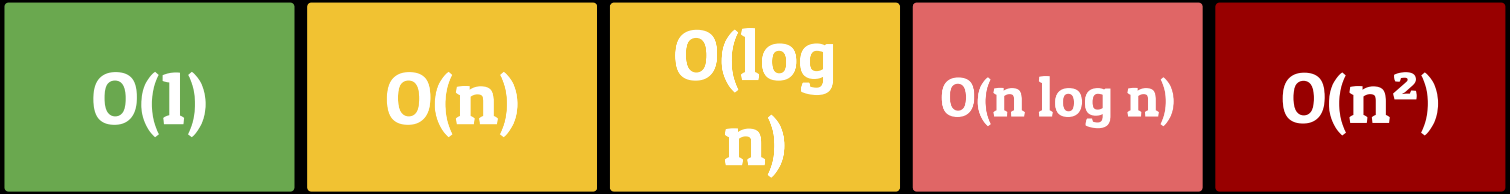
O(n)

O(log n)

O(n log n)

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations **mean** in terms of **efficiency**
  - 6 **most common** Time Complexity Equations



$O(1)$

$O(n)$

$O(\log n)$

$O(n \log n)$

$O(n^2)$

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- Let’s dive straight into what the actual equations **mean** in terms of **efficiency**
  - 6 **most common** Time Complexity Equations

$O(1)$ | $O(n)$ | $O(\log n)$ | $O(n \log n)$ | $O(n^2)$ | $O(2^n)$

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
    - No matter what the size of your data set is, the task will be completed in a single instruction

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can “score” on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

Size of Data Set (N)

1

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

Operations Required	Size of Data Set (N)
1	1

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
    - No matter what the size of your data set is, the task will be completed in a single instruction

Operations Required	Size of Data Set (N)
1	100

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

Operations Required	Size of Data Set (N)
1	1,000,000

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
    - No matter what the size of your data set is, the task will be completed in a single instruction

Operations Required	Size of Data Set (N)
1	800 Quadrillion

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

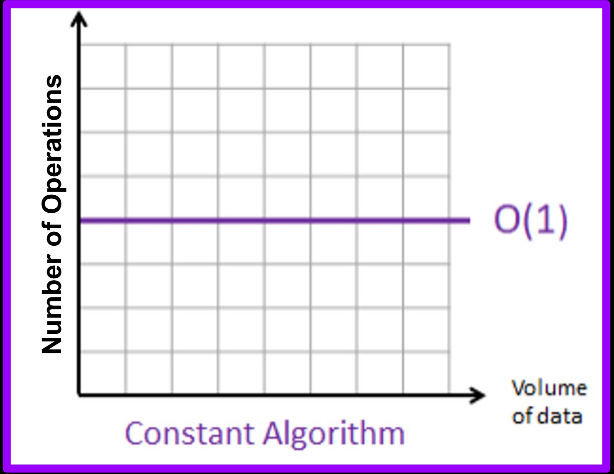
- The absolute best a data structure can “score” on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

When we graph Volume of Data vs # Of Instructions Required

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can “score” on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

When we graph Volume of Data vs # Of Instructions Required

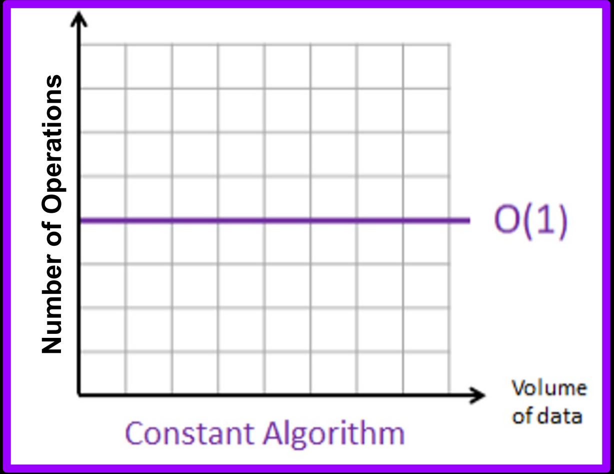


Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

When we graph Volume of Data vs # Of Instructions Required

# Of Operations remains constant

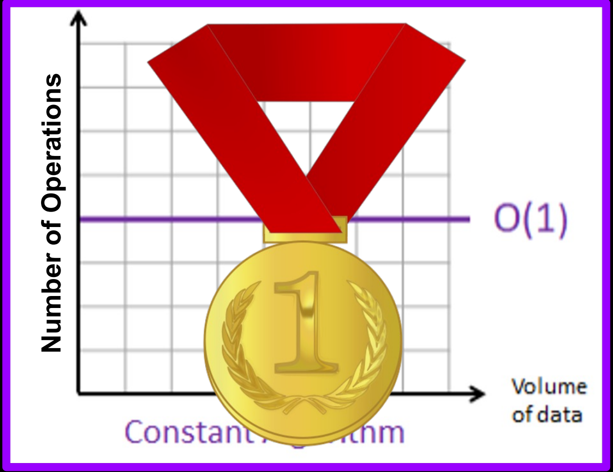


Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The absolute best a data structure can "score" on each criteria is O(1)
  - No matter what the size of your data set is, the task will be completed in a single instruction

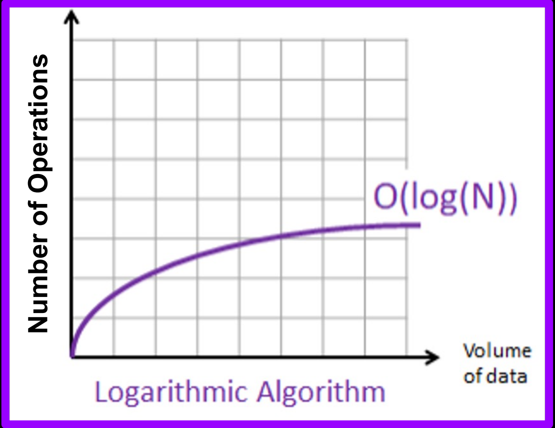
When we graph Volume of Data vs # Of Instructions Required

# Of Operations remains constant



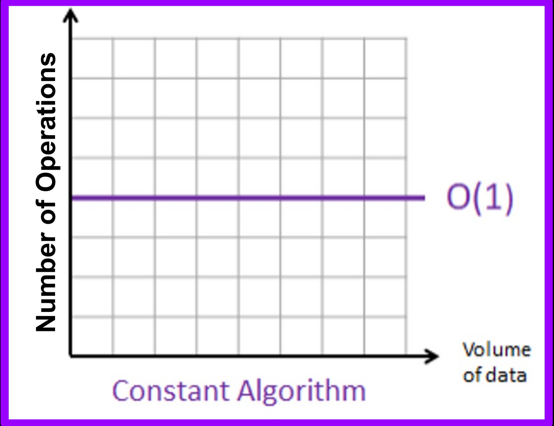
Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

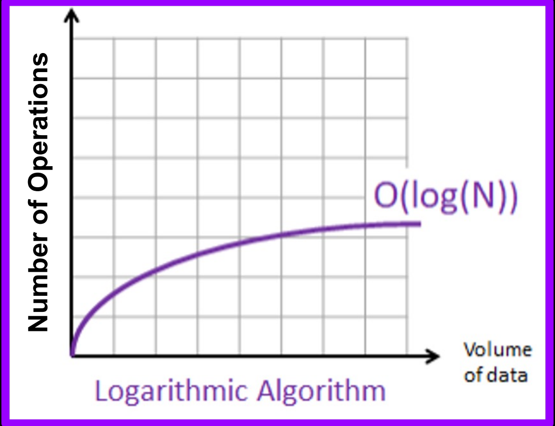
- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases

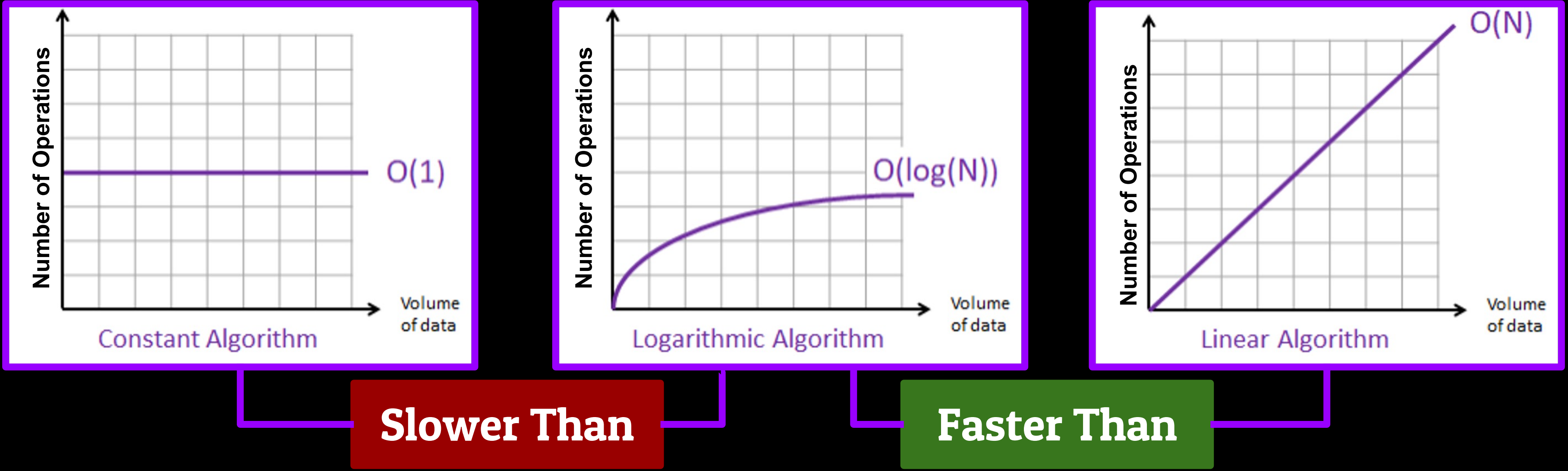




Slower Than

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next fastest type of time complexity equation os **O(log n)**
  - Still provides **fast** completion time
  - Gets **more efficient** as the size of the data set increases

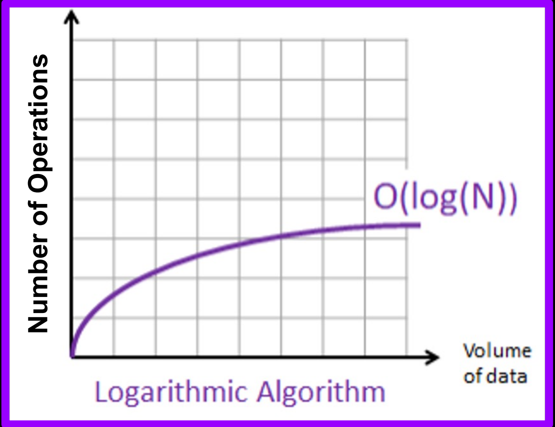


Slower Than

Faster Than

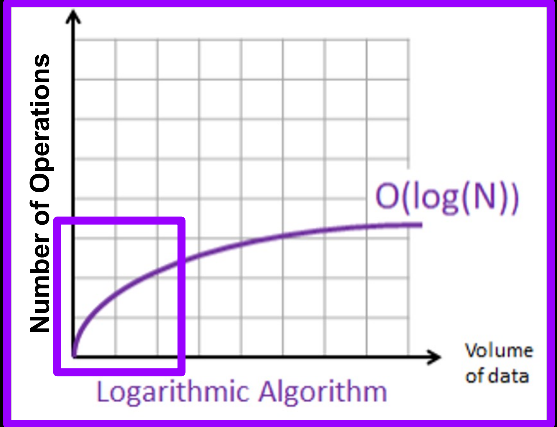
Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

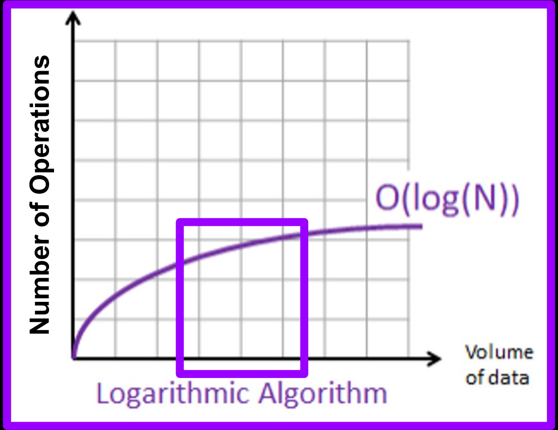
- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



Skyrockets

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

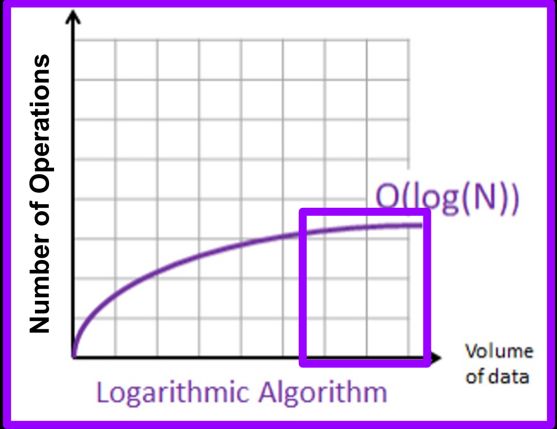
- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



Increases More Slowly

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

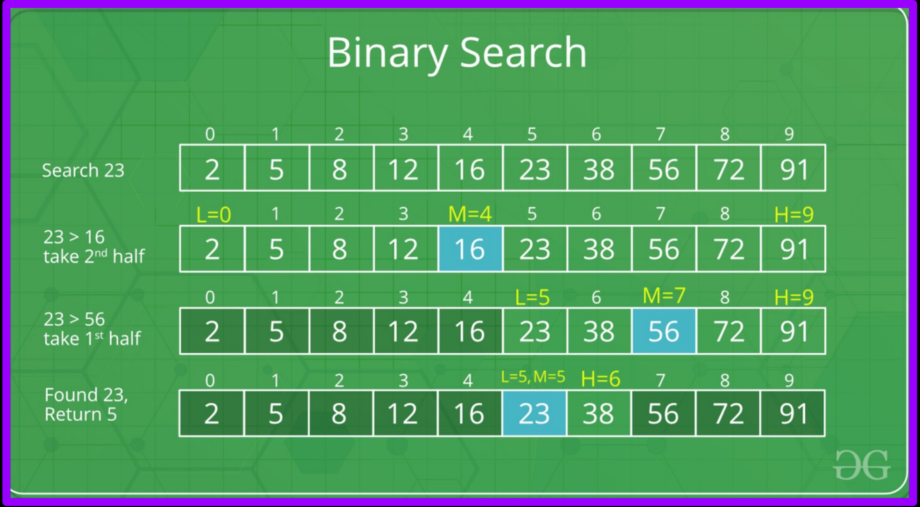
- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



Increases Way More Slowly

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases

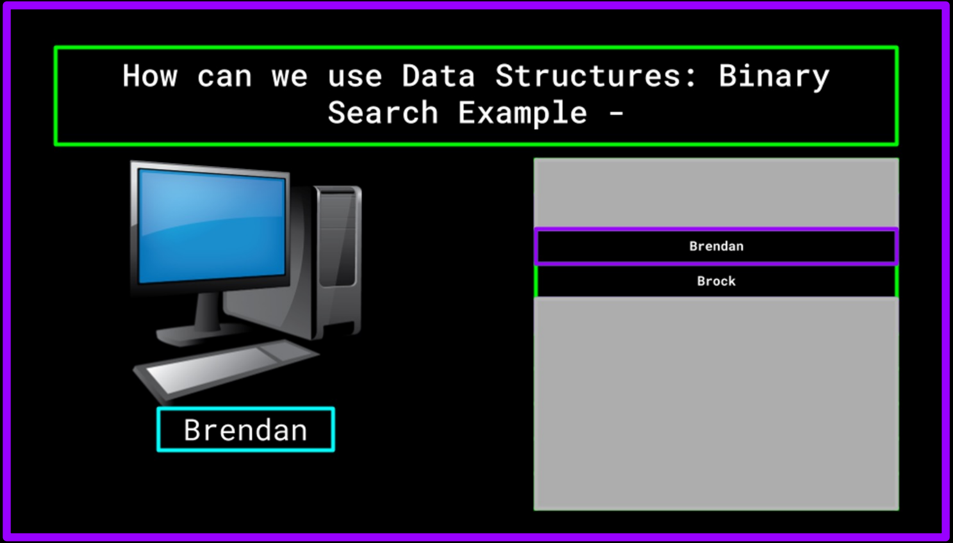


Binary Search

	0	1	2	3	4	5	6	7	8	9
Search 23	2	5	8	12	16	23	38	56	72	91
	L=0	1	2	3	M=4	5	6	7	8	H=9
23 > 16 take 2nd half	2	5	8	12	16	23	38	56	72	91
	0	1	2	3	4	L=5	6	M=7	8	H=9
23 > 56 take 1st half	2	5	8	12	16	23	38	56	72	91
	0	1	2	3	4	L=5, M=5	H=6	7	8	9
Found 23, Return 5	2	5	8	12	16	23	38	56	72	91

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next fastest type of time complexity equation os O(log n)
  - Still provides fast completion time
  - Gets more efficient as the size of the data set increases



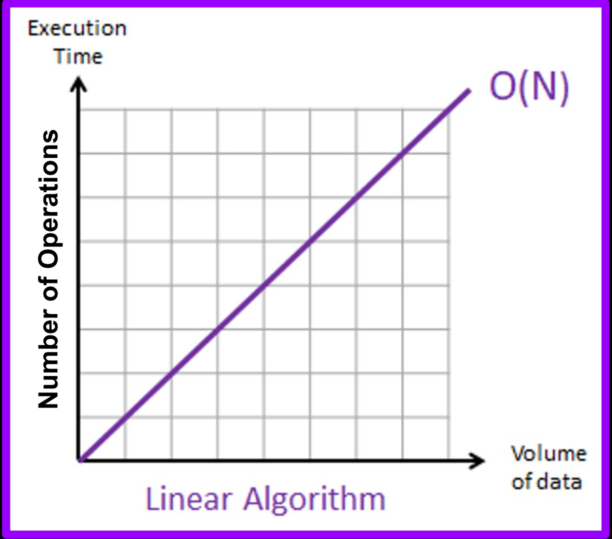
How can we use Data Structures: Binary Search Example -

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next common time complexity efficiency equation type that's going to come up is O(n)
  - The last of the "decent" equations

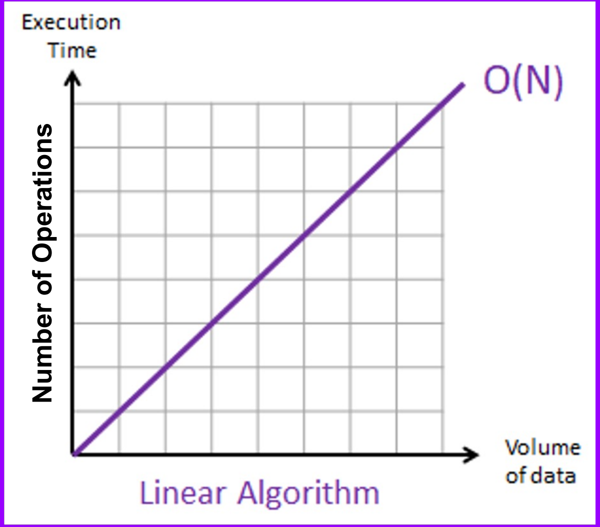
Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next common time complexity efficiency equation type that's going to come up is O(n)
  - The last of the "decent" equations



Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next common time complexity efficiency equation type that's going to come up is O(n)
  - The last of the "decent" equations

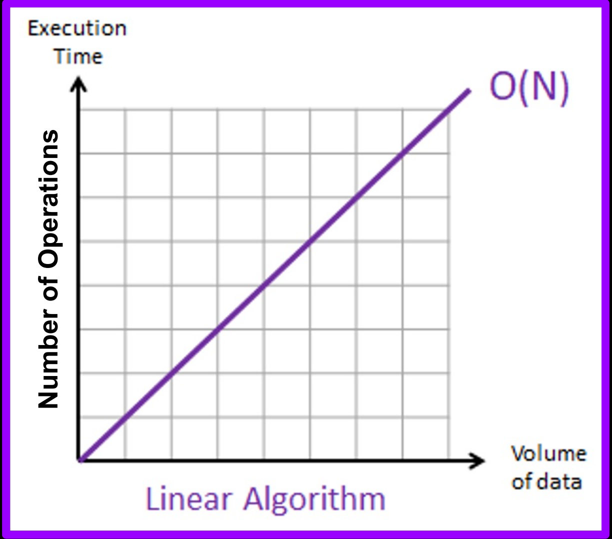


Size of Data Set (N)

10

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

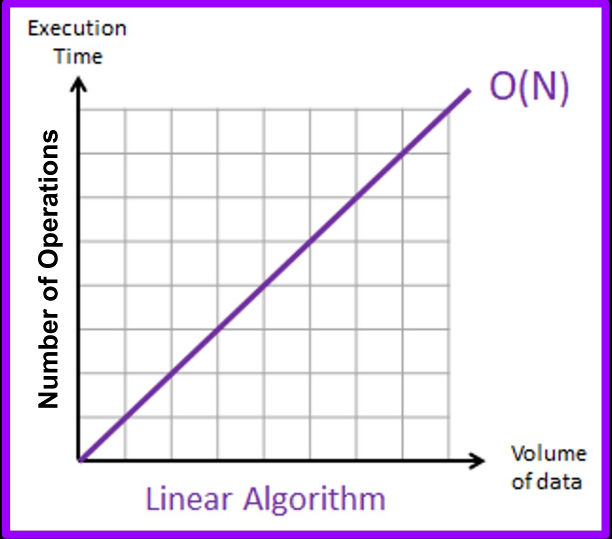
- The next common time complexity efficiency equation type that's going to come up is O(n)
  - The last of the "decent" equations



Operations Required	Size of Data Set (N)
10	10

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

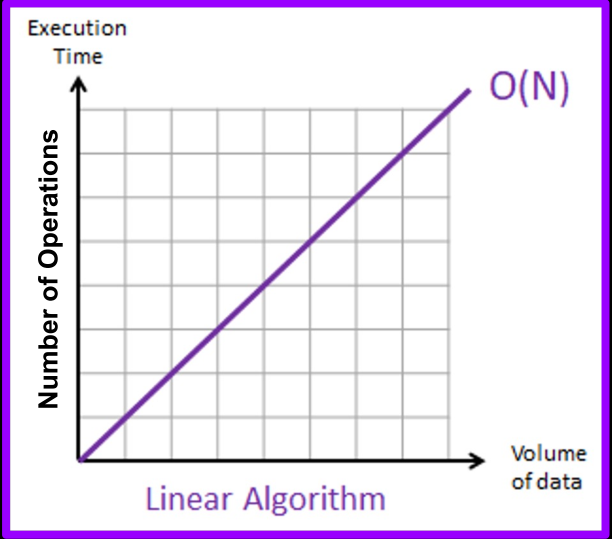
- The next common time complexity efficiency equation type that's going to come up is O(n)
  - The last of the "decent" equations



Operations Required	Size of Data Set (N)
50	50

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next common time complexity efficiency equation type that's going to come up is O(n)
    - The last of the "decent" equations



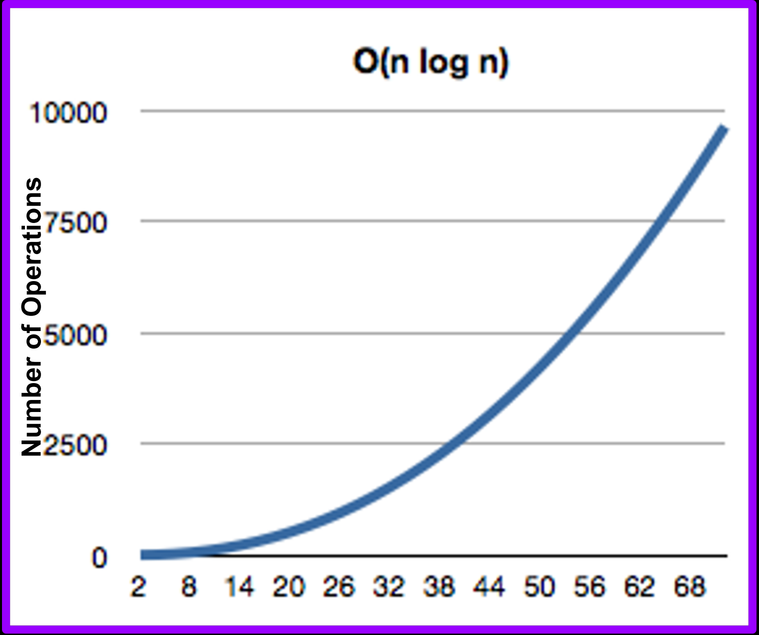
Operations Required	Size of Data Set (N)
1,000	1,000

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next type of equation that will come up is O(n log n)
    - The first which is relatively bad in terms of efficiency

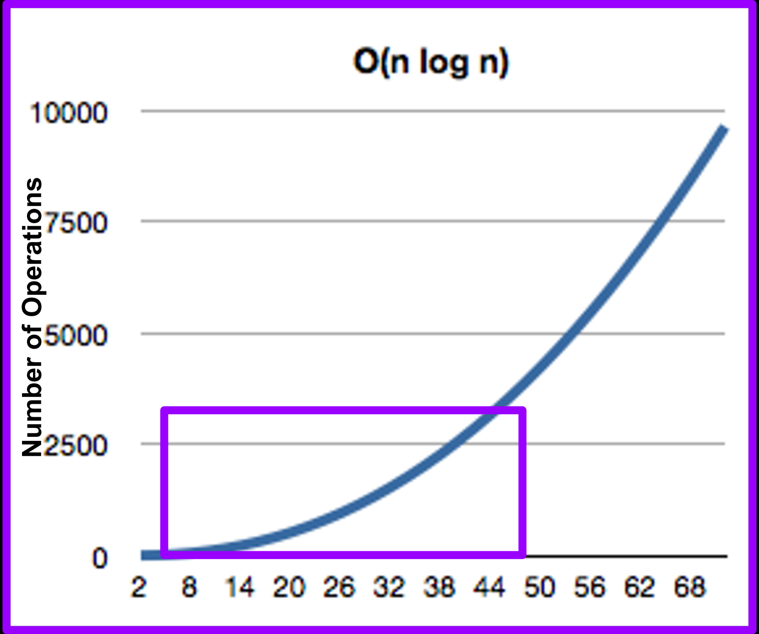
Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next type of equation that will come up is O(n log n)
  - The first which is relatively bad in terms of efficiency



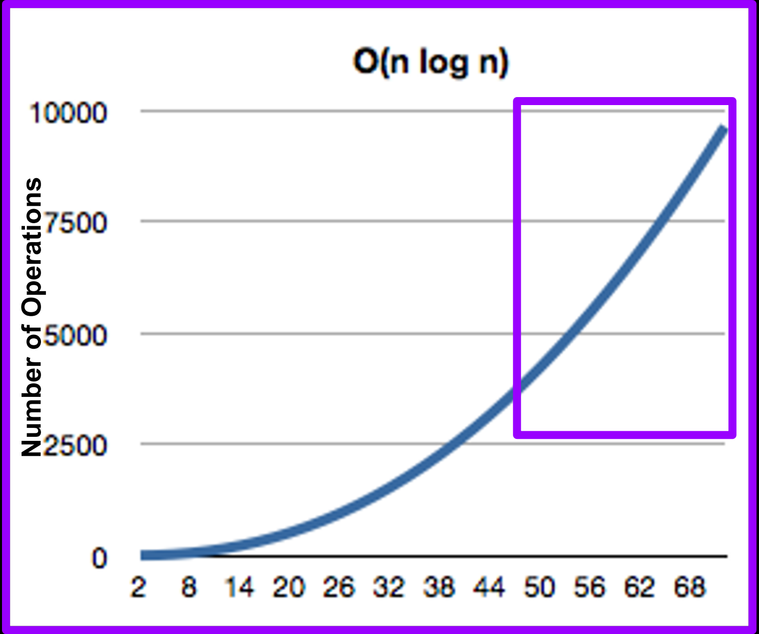
Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next type of equation that will come up is O(n log n)
  - The first which is relatively bad in terms of efficiency



Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The next type of equation that will come up is **O(n log n)**
  - The first which is **relatively bad** in terms of efficiency



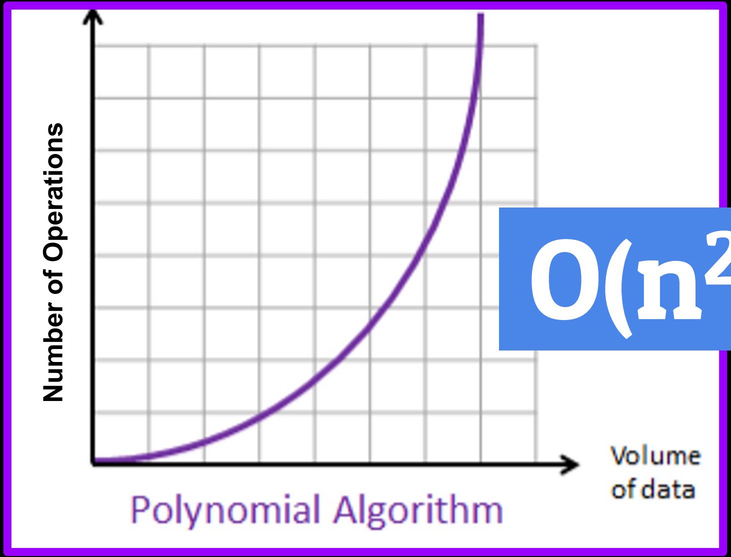
Slope Increases

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The last 2 types of equations are $O(n^2)$ and $O(2^n)$
    - Very bad in terms of efficiency
    - Exponential in structure

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

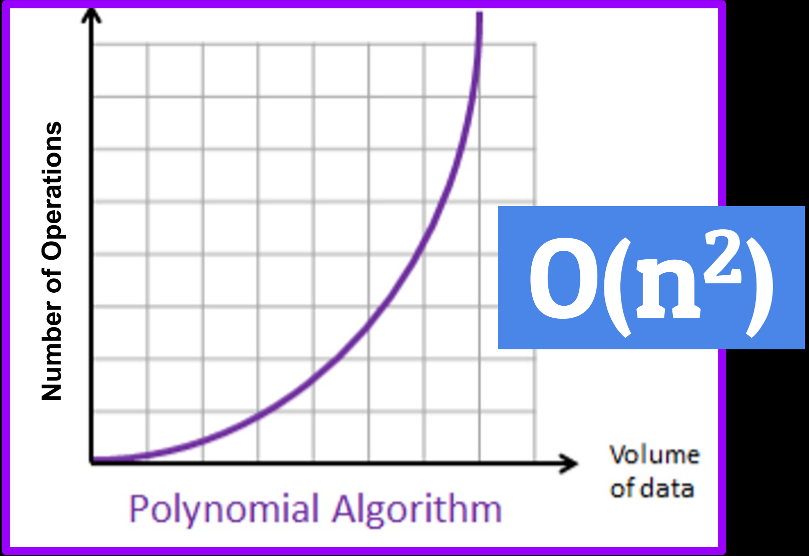
- The last 2 types of equations are $O(n^2)$ and $O(2^n)$
  - Very bad in terms of efficiency
  - Exponential in structure

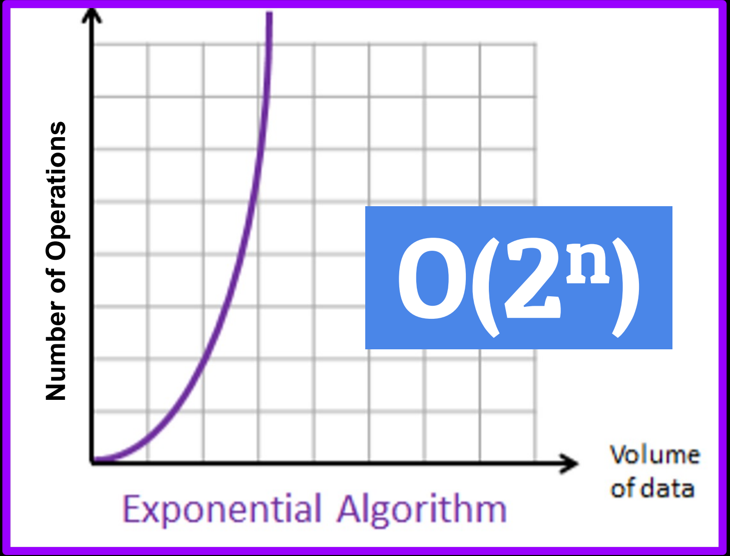


$O(n^2)$

Measuring Efficiency with BigO Notation - Types of Time Complexity Equations

- The last 2 types of equations are $O(n^2)$ and $O(2^n)$
  - Very bad in terms of efficiency
  - Exponential in structure





Measuring Efficiency with BigO Notation - Final Note on Time Complexity Equations

- Time Complexity Equations are **NOT** the only metric you should be using the gauge which data structure to use
    - Some provide other functionality that make them extremely useful

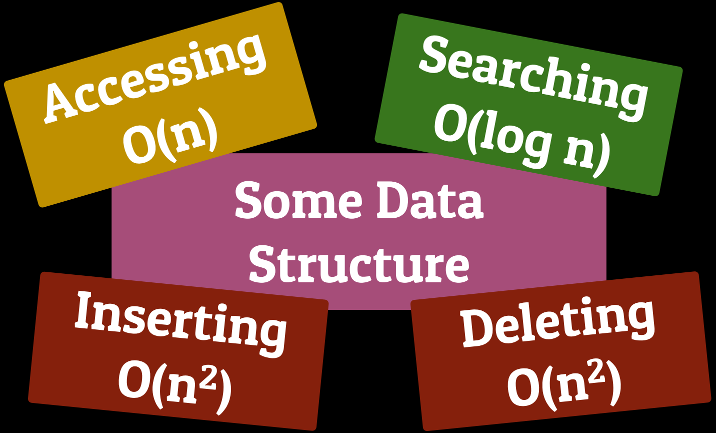
Measuring Efficiency with BigO Notation - Final Note on Time Complexity Equations

- Time Complexity Equations are **NOT** the only metric you should be using the gauge which data structure to use
    - Some provide other functionality that make them extremely useful

Some Data Structure

Measuring Efficiency with BigO Notation - Final Note on Time Complexity Equations

- Time Complexity Equations are **NOT** the only metric you should be using the gauge which data structure to use
    - Some provide **other functionality** that make them extremely useful



Accessing O(n)

Searching O(log n)

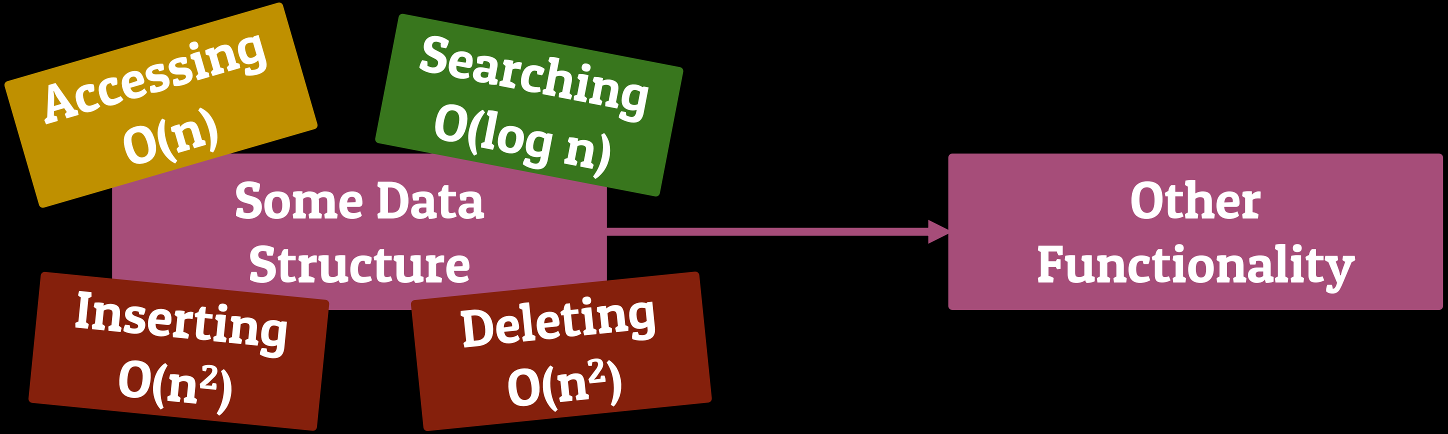
Some Data Structure

Inserting $O(n^2)$

Deleting $O(n^2)$

Measuring Efficiency with BigO Notation - Final Note on Time Complexity Equations

- Time Complexity Equations are **NOT** the only metric you should be using the gauge which data structure to use
  - Some provide **other functionality** that make them extremely useful


Accessing
O(n)
Searching
O(log n)
Some Data Structure
Inserting
O(n²)
Deleting
O(n²)
Other Functionality