MODULE 1: INTRODUCTION TO PROGRAMMING

Introduction to Collections (Part 2)

1 on 1s

Yesterday

- What is a **namespace**?
- What are some properties of a **list**?
- What are some properties of a **stack**?
- What are some properties of a **queue**?

Collections: Dictionary<T,T>

- A **dictionary** is an indexed collection that allows values to be located using user-defined keys.
- **You** define the data types of the key and values
- Keys **must** be unique

Accessing Elements in a Dictionary

Truck

Webster

a large, heavy motor vehicle used for transporting goods, materials, or troops.

Webster[“Truck”]

LET'S CODE!

Collections: HashSet<T>

- A **HashSet** is like a list except it does not allow duplicates.
- Elements are not kept in order

LET'S CODE!

Algorithmic Complexity

```
public bool IsLastElementEven(int[] array)
{
    return array[array.Length - 1] % 2 == 0;
}
```

We call this big O of O(1)

Big O Notation is a way to represent how long an algorithm will take to execute.

O(1) — Constant Time

- Constant time algorithms will always take same amount of time to be executed. The execution time of these algorithm is independent of the size of the input.

- Accessing a value with an array index.
- Push()
- Pop()

O(n) - Linear time complexity

- An algorithm has a linear time complexity if the time to execute the algorithm is directly proportional to the input size n. Therefore the time it will take to run the algorithm will increase proportionately as the size of input n increases.

```
for (int i = 0; i < nums.length; i++) {
  console.log(nums[i]);
}
```

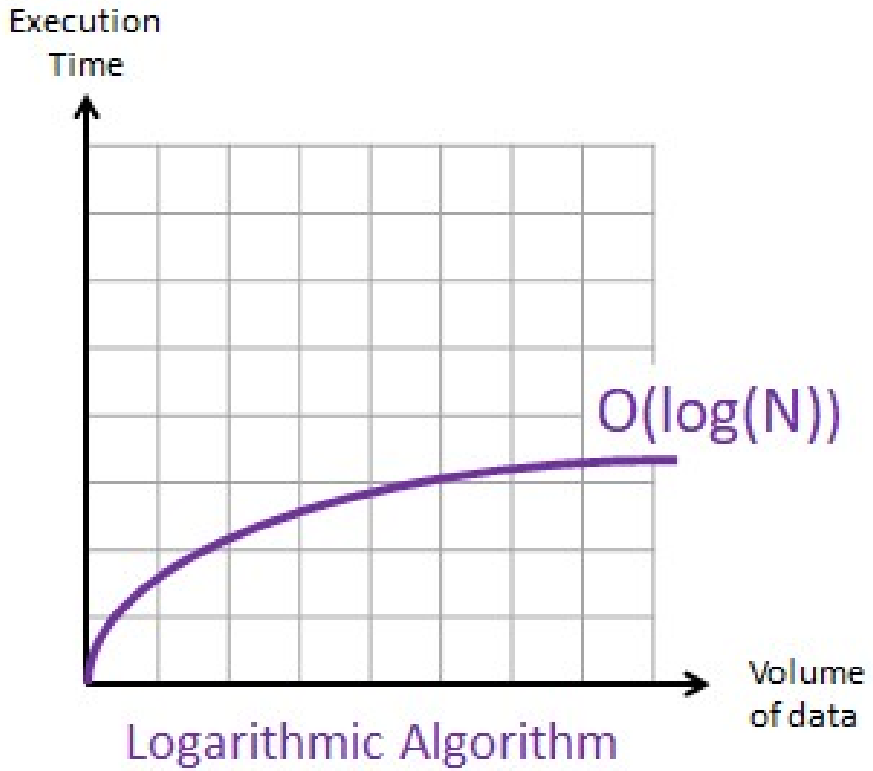
$O(n^2)$ - Quadratic time complexity

- An algorithm has quadratic time complexity if the time to execute it is proportional to the square of the input size.

```
for(var i = 0; i < length; i++) {     //has O(n) time complexity
    for(var j = 0; j < length; j++) { //has O(n^2) time complexity
      // More loops?
    }
}
```

O(log n) - Logarithmic time complexity

- An algorithm has logarithmic time complexity if the time it takes to run the algorithm is proportional to the logarithm of the input size n.



WHAT QUESTIONS DO YOU HAVE?

Reading for tonight:

Introduction to Classes