**Recursion**

• In your own words, what is recursion?

• Recursion is a function which calls itself and has a base case

• Why is it necessary to have a base case?

• Due to the nature of recursion being a function which calls itself, without a base case, a recursive function would never stop running

**Graphs**

What is a graph?

• A graph is a data structure made up of inter-related nodes. They contain loops, connected by arcs and can be non-directed.

How is a graph different from a tree?

• Bc unlike one or bi-directional trees, they can contain loops (“cycles”), and can be non-directed

Give an example of something that would be good to model with a graph.

• An ecosystem of animals and what they eat (bc they can eat each other or not, etc).

***Data Structure***

***Index***

***Search***

***Add-R***

***Add-L***

***Pop-L***

***Pop-R***

*Python List (Array)*

*O(1)*

*O(n)*

*O(1)*

***O(n)***

***O(1)***

***O(n)***

*Linked List*

***O(n)***

***O(n)***

***O(1)***

***O(1)***

***O(1)***

***O(n)***

*Doubly-Linked List*

*O(n)*

*O(n)*

*O(1)*

*O(1)*

*O(1)*

*O(1)*

*Queue (as Array)*

*X*

*X*

*O(1)*

*X*

*O(n)*

*X*

*Queue (as LL or DLL)*

*X*

*X*

*O(1)*

*X*

*O(1)*

*X*

*Stack (as Array, LL, or DLL)*

*X*

*X*

*O(1)*

*X*

*X*

*O(1)*

*Deque (as DLL)*

*O(1)*

*O(1)*

*O(1)*

*O(1)*

***Data Structure***

***Get***

***Add***

***Delete***

***Iterate***

***Memory***

*Dictionary (Hash Map)*

*O(1)*

*O(1)*

*O(1)*

*O(n)*

*medium*

*Set (Hash Map)*

*O(1)*

*O(1)*

*O(1)*

*O(n)*

*medium*

*Binary Search Tree*

*O(log n)*

*O(log n)*

*O(log n)*

*O(log n)*

*Low*

*Tree*

*O(n)*

*O(n)*

*O(n)*

*O(n)*

*Low*

**Sorting**

• Describe in words how the Bubble Sort algorithm works.

You compare each item with the one next to it, advancing each object with the comparatively higher value until it bubbles to the top of the line. You then repeat the same with each subsequent position.

• Describe in words how the Merge Sort algorithm works.

You take your data and break it down into lists of one item each, comparing each item to the one next to it, eventually ending up with 2 sorted lists. You then merge those sorted lists together by comparing the first item in each list against each other.

• Describe in words how the Quick Sort algorithm works.

Arranging numbers to either side of a pivot number.