Week 3 Data Structures

Algo Efficiency/ Big O

Growth Rates by comparing their execution time of two algorithms.

Big O

Exec time/memory, input size and execution time

Time is doubled when the number of inputs is doubled, an algoriithm grows at a linear rate

Formal Definition of BIG 0

Growth rate will be determined by growth rate of fastet growing term(largest exponent)

Safe to ignore all constants and drop the lower order terms when determining the order.

O(1) Constant

O(Log n) Logarithmic

O(n) linear

O(n log n) log-linear

O(n^2) Quadratic

O(n^3) Cubic

O(2^n) Exponential

O(n!) Factorial

A table with numbers and symbols

Description automatically generated

Array List

Array is indexed data structure

Things u cant do with array :

* Increase decrease length, fixed
* Add element at specific position without shifting the other elements
* Remove an element from specific postiion without shifting other elements

List Interface

A black and white text on a white background

Description automatically generated

List

Return a reference to an element at a specified location (method get)

• Find a specified target value (method get)

• Add an element at the end of the list (method add)

• Insert an element anywhere in the list (method add)

• Remove an element (method remove)

• Replace an element in the list with another (method set)

• Return the size of the list (method size)

• Sequentially access all the list elements without having to manipulate a subscript.

ArrayList

Size is automatically increased and decreases as elements are removed

Array list has instance to return its current size

Array list has a capacity

The statements:

• declare List variables • add()

• size()

• remove()

• get()

• set()

• indexOf()