Introduction to Common Data Structures

Learn How to Organise Data in Your Software Designs

Boniface Kabaso

November 8, 2017

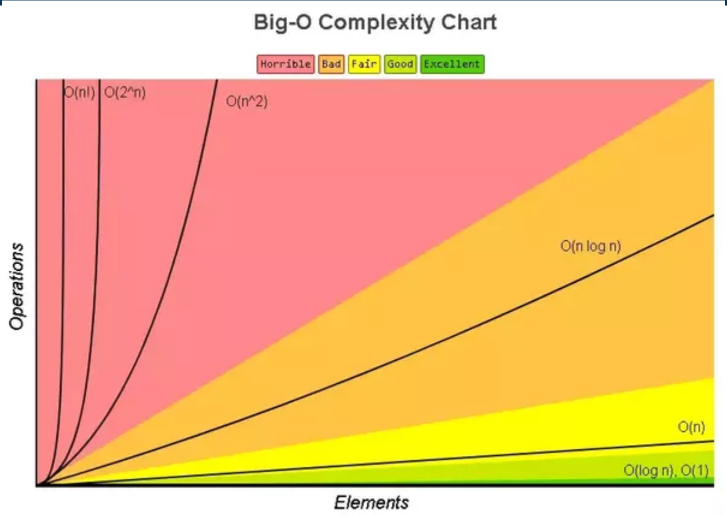
Outline

Outline

Outline

Explanation

Best Performance



Time Complexity
Constant Time: O(1)

An algorithm is said to run in constant time regardless of the input size.

1. Accessing element of an array
2. Push and pop of a stack.
3. Enqueue and remove of a queue.
4. Accessing an element of Hash-Table.
5. Bucket sort

Time Complexity
Linear Time: O(n)

An algorithm is said to run in linear time if the execution time of the algorithm is directly proportional to the input size.

1. Array operations like search element, find min, find max etc.
2. Linked list operations like traversal, find min, find max etc.

Time Complexity
Logarithmic Time: O(logn)

An algorithm is said to run in logarithmic time if the execution time of the algorithm is proportional to the logarithm of the input size. Each step of an algorithm, a significant portion of the input is pruned out without traversing it. of the input size.

1. Binary Search

Time Complexity

N-LogN Time: O(nlog(n))

An algorithm is said to run in logarithmic time if the execution time of an algorithm is proportional to the product of input size and logarithm of the input size.

1. Merge Sort
2. Quick Sort
3. Heap Sort

Time Complexity

Quadratic Time: O(n power x)

An algorithm is said to run in logarithmic time if the execution time of an algorithm is proportional to the square of the input size.

1. Bubble Sort
2. Selection Sort
3. Insertion Sort

Time Complexity
Getting Run Time

Constants
Each statement takes a constant time to run. Time Complexity is O(1)

Loops
The running time of a loop is a product of running time of the statement inside a loop and number of iterations in the loop. Time Complexity is O(n)

Nested Loops
The running time of a nested loop is a product of running time of the statements inside loop multiplied by a product of the size of all the loops. Time Complexity is O(n power x)

Time Complexity
Getting Run Time

Consecutive Statements

Just add the running times of all the consecutive statements

If-Else Statements

Consider the running time of the larger of if block or else block. And ignore the other one.

Logarithmic Statement

If each iteration the input size is decreases by a constant factors. O(log n)

Outline

Outline

Data Structure Types

Features of OO languages

Any Language that claims to be Object Oriented must at least pass the test on these three Features

1. Arrays
2. LinkedList
3. Stacks
4. Queues

Outline

Arrays

Linear Data Structures

An array is a list of values basied on a particular type, eg ints, String, Objects, etc. Each value is stored at a specific, numbered position in the array. The number corresponding to each position is called an index or a subscript.

Listing 1: Code Sample for Arrays

```
     Student a = new Student();
     Student b = new Student();
     Student c = new Student();
    //Initialize tohold three students
    Student[] studentArray = new Student[3];
   //Set values
   studentArray[0]=a;
   studentArray[1]=b;
   studentArray[2]=c;
```

Outline

LinkedList
Linear Data Structures

The linked list is a list of items, called nodes. Nodes have two parts, value part and link part. Value part is used to stores the data. The value part of the node can be either a basic data-type like an integer or it can be some other data-type like an object of some class. The link part is a reference, which is used to store addresses of the next element in the list.

Listing 2: Code Sample for Linked List

```
//Array backed lists
List a = new ArrayList();
//Linked lists
LinkedList l = new LinkedList();
```

Outline

Stack
Linear Data Structures

A stack is an abstract data structure that serves as a collection of objects that are inserted and removed based on a last-in first-out (LIFO) principle. Accordingly, the two operations that most clearly define a stack structure are push, which adds objects to the collection, and pop, which removes objects from the collection.

Outline

Stack
Linear Data Structures

Java provides a concrete implementation of the stack data structure through the Stack¡T¿ generic class.

Listing 3: Code Sample for Stack

```java
Stack s = new Stack();
s.push(anObject);
s.pop();
```

Outline

Stack
Linear Data Structures

A queue is an abstract data structure that serves as a linear collection of objects that are inserted and removed based on the first-in first-out (FIFO) principle. The two most notable operations of a queue are enqueue, which adds objects to the tail or back of the collection, and dequeue, which removes objects from the head or front of the collection.
Java provides the abstract Queue¡E¿ interface, and several concrete implementations of the queue data structure use this interface.

Outline

Stack
Linear Data Structures

Queue is also extended to the Deque¡E¿ interface that represents a double-ended queue. The ArrayDeque¡E¿ class is one concrete implementation of the Deque¡E¿ interface:

Listing 4: Code Sample for Arrays

```
ArrayDeque q = new ArrayDeque();
q.addLast(anObject);
q.getFirst();
```

Industry Day

Panel Discussion

Questions ?

Boniface Kabaso
kabasoB AT cput DOT ac DOT za