PSLG

Week 07

Dawid Sobczak & Tomasz Zajas

1. Plan

2. Data Structures

3. Big O

4. Command Line Arguments

Plan

- Overview of data structures and Big O.
- Do some lab questions / Cover any material ye need help with.

What is a Data Structure?

Definition

A system of organizing data in software that allows more optimal searching, categorizing, or storage of information.

Examples

Linked lists, arrays e.t.c.

Why use Data Structures?

Reason #1

Storing, managing and manipulating complex sets of data can be a burden on the programmer. A higher level abstraction like a data structure can help keep things simple when working at a higher level.

Example

Manipulating an image through a set of functions instead of through a multi-dimensional array.

Why use Data Structures?

Reason #2

There are many common patterns and activities in software. Algorithms and data structures are general utilities that can be used to solve a lot of common problems.

Example

Sorting algorithm + Searching algorithm to get better speeds when searching for an item in a list.

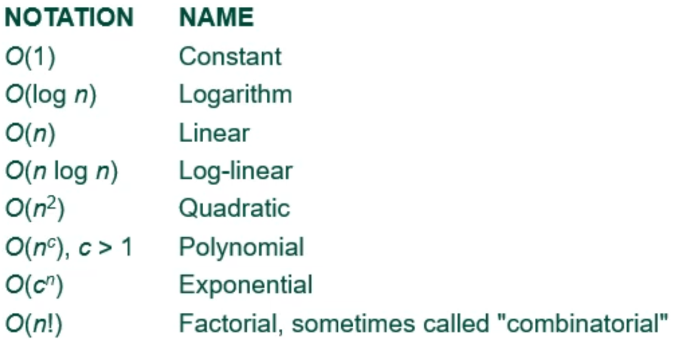
Big O Notation

Big O notation is used to categorize algorithms by their performance when given a very large input. For example, in order to sort an array of 100 elements:

- Bubble sort, with $O(n^2)$ will take $100^2 \Rightarrow 10,000$ units of time.
- Quick sort, with $O(n \cdot \log 100)$ will take $n \cdot \log(100) \Rightarrow 200$ units of time.

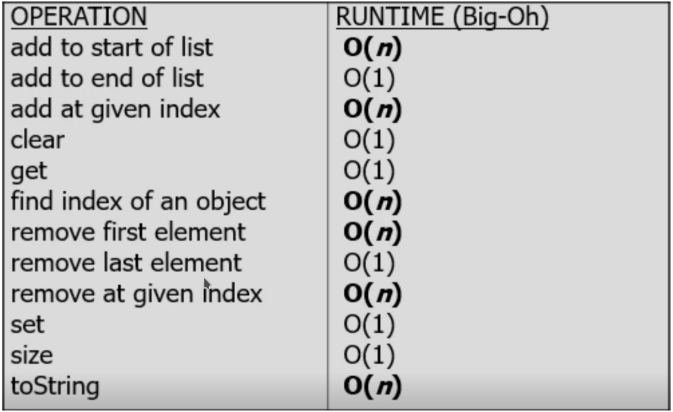
Big O notation signifies that the algorithm will never take longer than O (<whatever>) length of time.

Common Big O functions



NOTATION	NAME
$O(1)$	Constant
$O(\log n)$	Logarithm
$O(n)$	Linear
$O(n \log n)$	Log-linear
$O(n^2)$	Quadratic
$O(n^c)$, $c > 1$	Polynomial
$O(c^n)$	Exponential
$O(n!)$	Factorial, sometimes called "combinatorial"

Array List Functions



OPERATION	RUNTIME (Big-Oh)
add to start of list	**O(*n*)**
add to end of list	O(1)
add at given index	**O(*n*)**
clear	O(1)
get	O(1)
find index of an object	**O(*n*)**
remove first element	**O(*n*)**
remove last element	O(1)
remove at given index	**O(*n*)**
set	O(1)
size	O(1)
toString	**O(*n*)**

Command Line Arguments

1. Users can pass the arguments during the execution by passing the command-line arguments inside the main () method.
2. The arguments passed from the console can be received in the java program and it can be used as an input.
3. You can pass in any parameters such as String, double, int etc.
4. The arguments are converted into String and passed into the String args[] array declared in the main function.

Sample Programs

Sample Program #1

```
class CommandLineExample {
    public static void main(String args[]) {
        System.out.println("Your first argument is: " + args[0]);
    }
}
```

Sample Program

Sample Program #2

```
class CommandLineExample {
    public static void main(String args[]) {
        for (int i = 0; i < args.length; i++) {
            System.out.println(args[i]);
        }
    }
}
```