

**FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHONOLOGY**

[***data***](http://learninghub.upm.edu.my/blastdk/course/view.php?id=3428) ***structures and algorithms***

**SSK3118**

**GROUP 3**

***Bank Queue Simulation***

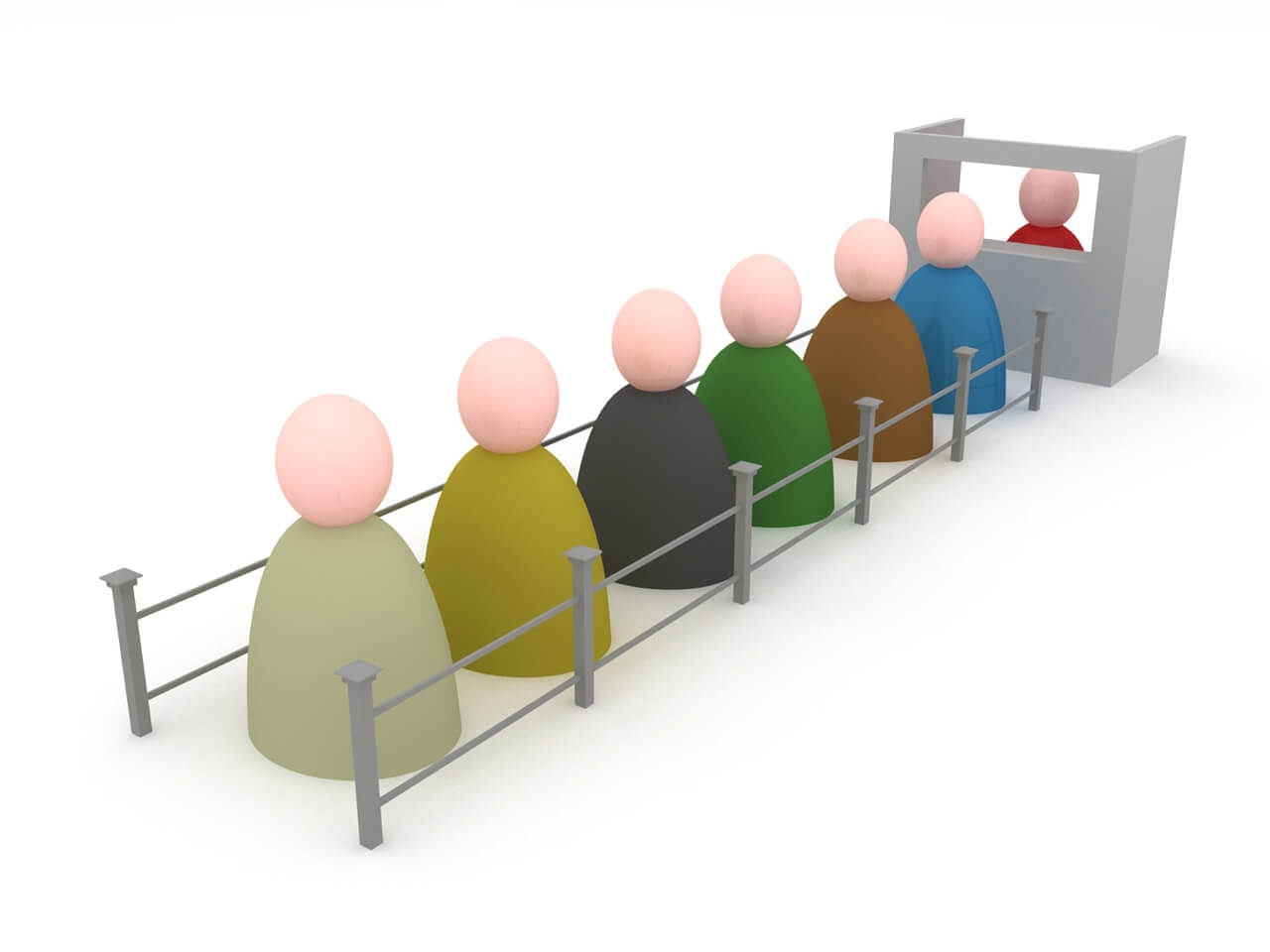
**Prepared for:**

**Dr. Mohd Hafeez Bin Osman**

**Prepared by:**

|  |  |
| --- | --- |
| **Name** | **Matric No.** |
| Md Shohiduzzaman Shakil | 195674 |
| Gulam Mohiuddin Abdul Qader | 195461 |

**Bank Simulation**



**Introduction**

This program simulates a bank with multiple servers and customers. A bank service area consists of several servers and customer queues. If servers are busy, customers in the queue must wait for a server. If a server is free and customers are waiting, the first customer in the queue advances to the server's counter and begins their transaction. When a customer is done, they depart and the server becomes free. The simulation is run through many units of time. At the end of each time unit, the program prints out a snapshot of the queues, customers, servers, interval time of customer arrival and service time per customer. The program ends with printing out statistics of the simulation.

**The limits of simulation parameters**

* Maximum number of servers 10.
* The estimated average customer per hour is 100.
* The estimated interval time of customer arrival is between 30-45 seconds (random).
* The estimated service time per customer is between 120-600 seconds (random).

**Queue**

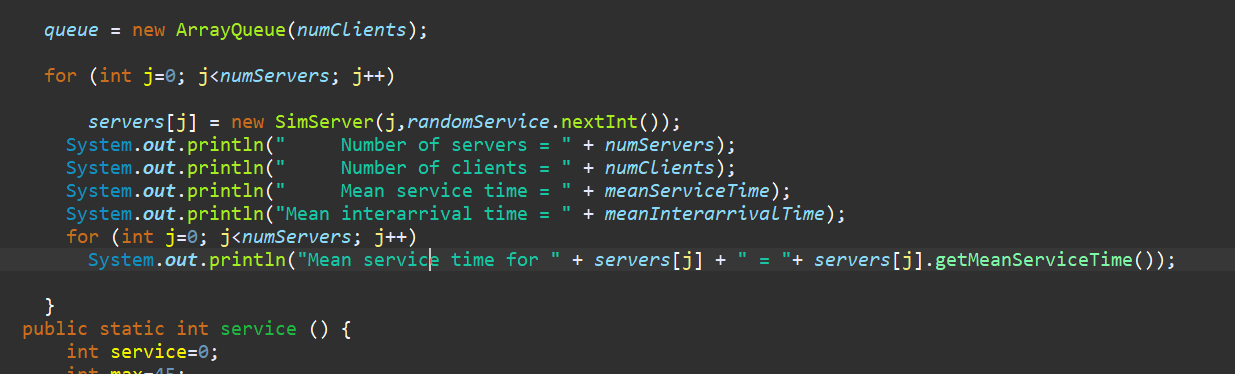


Queue is a linear data structure which enables the user to add elements from the rear end and remove elements from the front end only, using the concept of FIFO. FIFO means “First In First Out”.

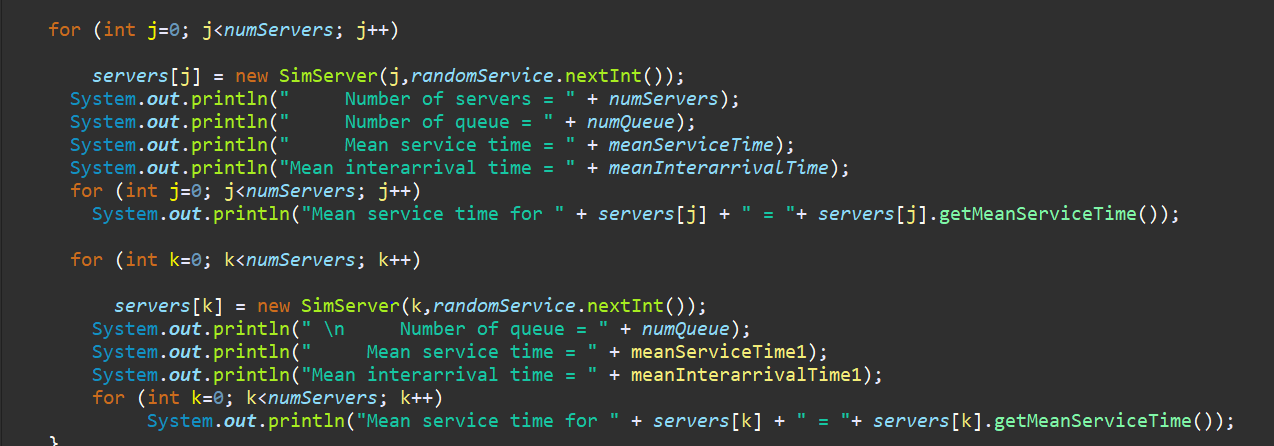
|  |  |  |
| --- | --- | --- |
| Summary of Queue methods | | |
|  | *Throws exception* | *Returns special value* |
| **Insert** | [add(e)](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#add-E-) | [size()](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#offer-E-) |
| **Remove** | [remove()](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#remove--) | [poll()](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#poll--) |
| **Examine** | [element()](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#element--) | [peek()](https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html#peek--) |

* **add()-** This method is used to add elements at the tail of queue. More specifically, at the last of linked-list if it is used, or according to the priority in case of priority queue implementation.
* **peek()-** This method is used to view the head of queue without removing it. It returns Null if the queue is empty.
* **element()-** This method is similar to peek(). It throws *NoSuchElementException* when the queue is empty.
* **remove()-** This method removes and returns the head of the queue. It throws *NoSuchElementException* when the queue is empty.
* **poll()-** This method removes and returns the head of the queue. It returns null if the queue is empty.
* **size()-** This method return the no. of elements in the queue.

**Big O Notation**

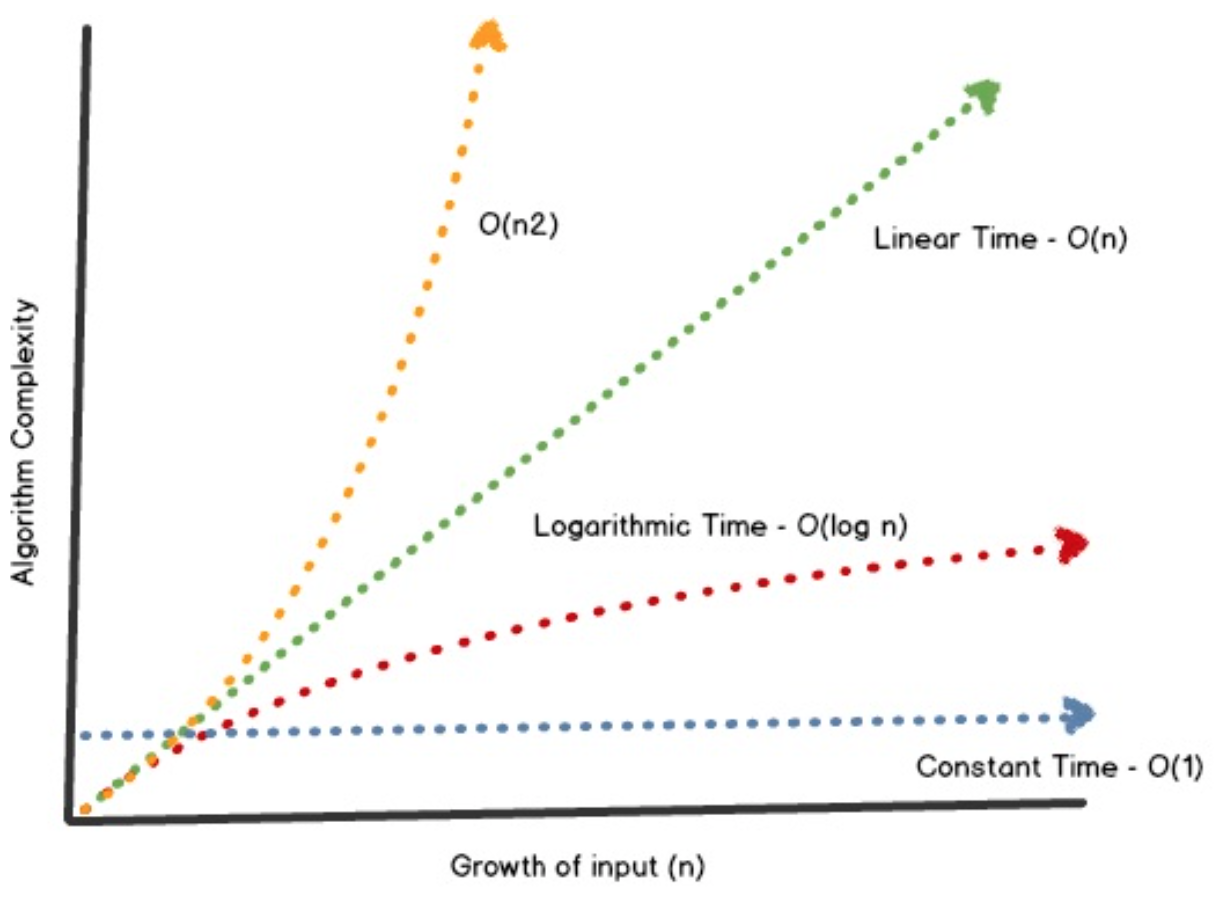


Here, there are 2 loops in the program but none of them depends on another. So the big O of this program will be, O(n)=4n.



Here, there are 4 loops in the program but none of them depends on another. So the big O of this program will be, O(n)=8n.

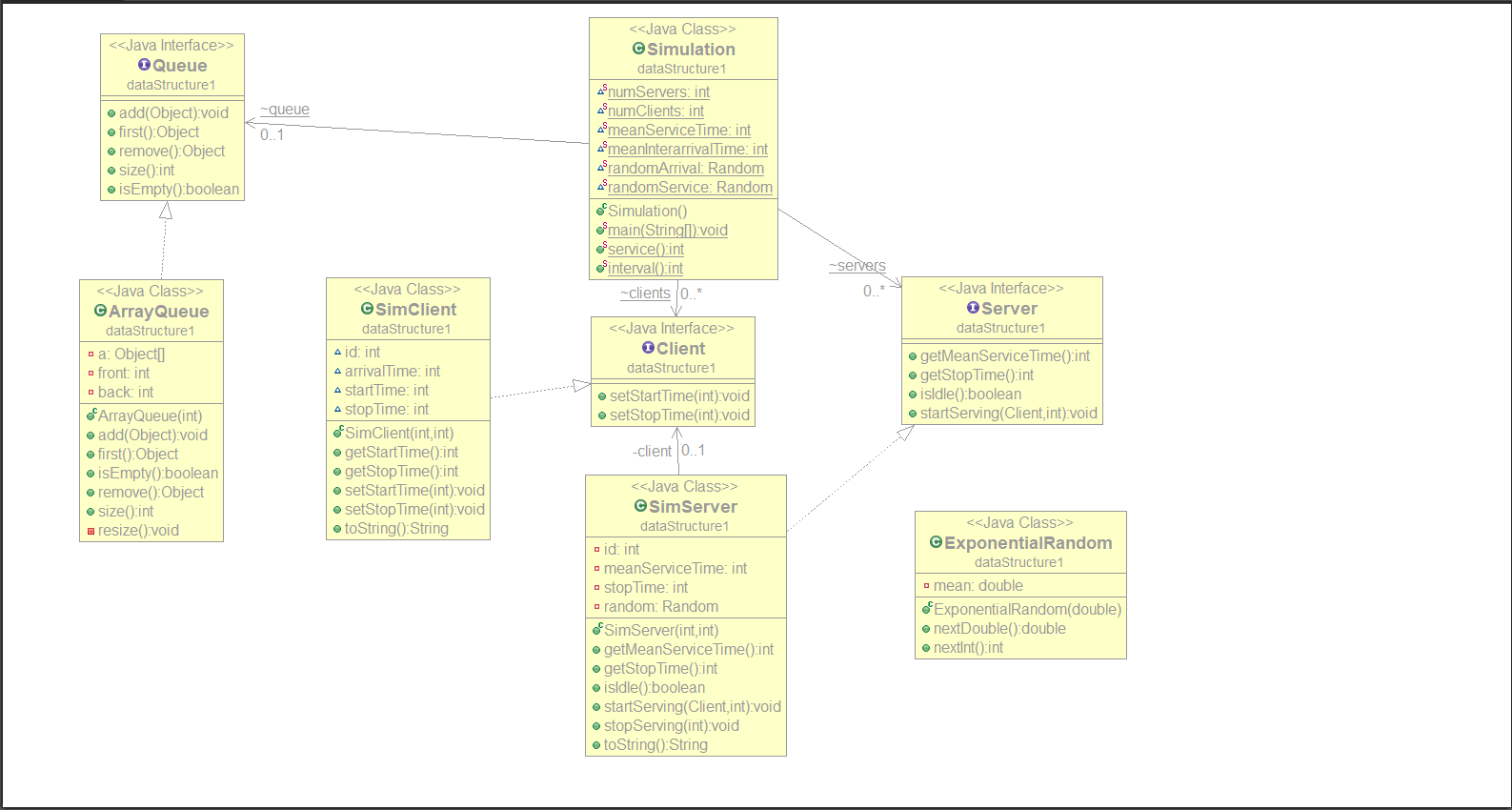
**Comparison Between Number Of Servers**

****

As we have linear time for both of the servers. An algorithm with linear complexity will take more time, by a constant gradient, as we give it more items to process. The larger the *n* value, the longer it takes, with the time being *n* times larger than if *n* = 1. So, from here we can safely say 4n will process faster than 8n. In simple words, using one queue with two more sever will make the simulation faster than using many queue with many server.

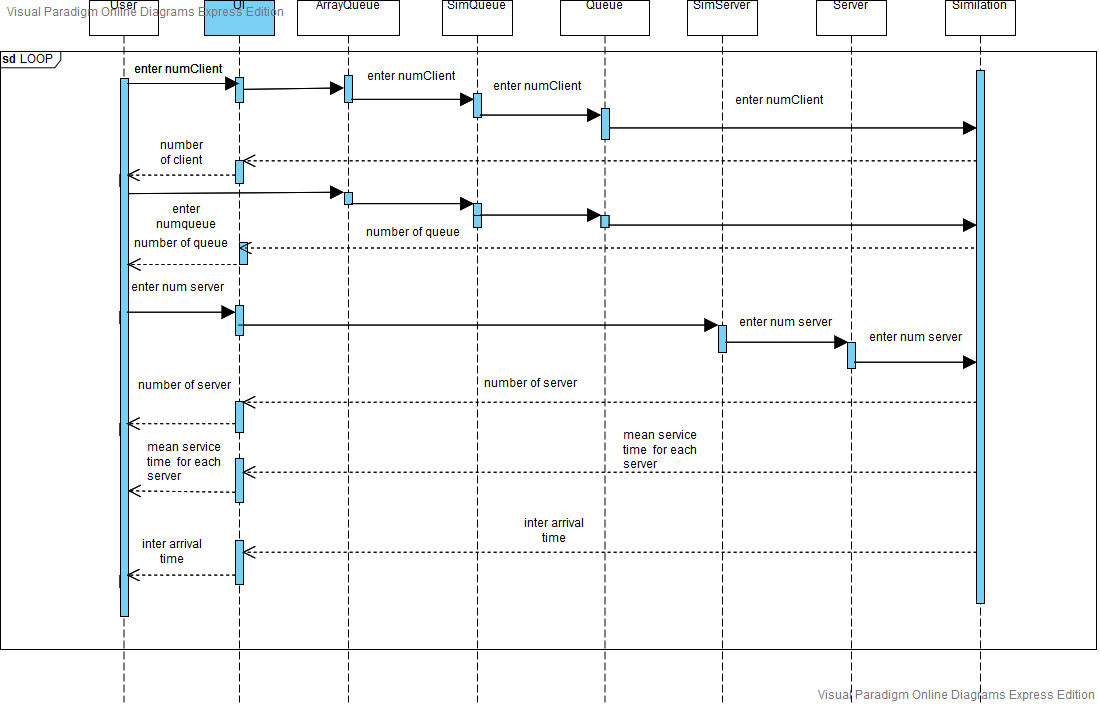
**Design**

**Class Diagram**



A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Sequence Diagram



A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

**Conclusion**

In conclusion, we can say that one queue with two or more server is best for the simulation over many server with many queue.

**References-**

<https://docs.oracle.com/javase/8/docs/api/java/util/Queue.html>

<https://www.quora.com/What-is-queue-in-Java-programming>

<https://en.wikipedia.org/wiki/Class_diagram>

<https://en.wikipedia.org/wiki/Sequence_diagram>

<https://towardsdatascience.com/linear-time-vs-logarithmic-time-big-o-notation-6ef4227051fb>

<https://www.trccompsci.online/mediawiki/index.php/Comparing_Algorithms_-_Big_O>