Computer Science & Information Systems

**System for Data Analytics – Lab Sheet 1**

**LOR – Locality of Reference**

1. Objective:

Students should be able to

1. Understand the memory organization and Principle of LOR (Locality of Reference)
2. Get hands-on experience to understand sequential access and random access in memory

Data Inference: Programs tend to reuse data and instructions they have used recently.

**90/10 rule**comes from empirical observation:  
*"A program spends 90% of its time in 10% of its code“*

*Implication:* predict with reasonable accuracy what *instructions* and *data* a program will use in the near future based on the recent past

**Principle of Locality of Reference (LOR)**

* + Program access **a relatively small amount of data** that is, a small amount of the address space or memory references **at any instant of time during execution**
  + Locality is a property of programs which is exploited in hardware design
  + **LOR in Time – Temporal Locality**
    - If a data item is referenced in the program, it will tend to be referenced again soon
  + Examples
    - Instructions in the body of a loop
    - Parameters / Local variables of a function / procedure
    - Data item (or a variable) that is computed iteratively, e.g. a cumulative sum or product
  + **LOR in Space – Spatial Locality**
    - If a data item is referenced in a program, data items whose addresses are close by tend to be referenced soon
  + Examples
    - Linear sequences of instructions (straight line code)
    - Elements of Arrays (accessed sequentially)

1. Steps to be performed:

Tools required: JDK 1.8, Any Java IDE (Eclipse or Netbeans)

Run 1:

* 1. Create a new project in Eclipse or Netbeans
  2. Create a source program ‘ArrayList.java’ in the project workspace
  3. Run and check the output
  4. Vary the value of ‘iterations’ variable and note down the time taken

Run 2:

a. Create a new source program ‘LinkedList.java’ in the project workspace

b. Run and check the output

c. Vary the value of ‘iterations’ variable and note down the time taken

Iterations – 1000, 10000, 100000, 1000000, 10000000

1. Outputs/Results:

* Students should try to plot a graph Time taken vs Iterations for both the runs – ArrayList and LinkedList.

1. Observations:

* Students need to observe the difference in the time taken for arraylist and linkedlist due to different access mechanism used by both the programs in memory w.r.t LOR.
* Why arraylist faster when compared to linkedlist for very large number of memory access?

1. References: