**DESIGN AND ANALYSIS OF ALGORITHMS**

**School name: Virginia International University**

**Course Name and Number: Design and Analysis of Algorithms, CMP 561 A**

**Date: 04-19-2015**

**Assignment Reference: Project**

**GROUP MEMBERS:**

1. **CHANDRA KIRAN GUNDU ( SID:10000128310)**
2. **SAI KRISHNA AMMISETTI (SID:10000127144)**
3. **SHARVANI RACHARLA (SID:10000123867)**
4. **VIJAYA LAKSHMI DADI (SID:10000127952)**
5. **SUCHETHA REDDY ANKSAPUR (SID:10000127600)**
6. **VENKATESWARA REDDY(SID: 10000128888)**

**Team Project**

**Scenario:**

The scenario we use to compare the CPU efficiency and memory efficiency among the data structure isto maintain a data table of employee salary details

1. The given data file “finalprojectdata.csv” stores the details of employees like their ID, AGE, SALARY

The finalprojectdata.csv has the fields: Id, age, salary

Each of them separated by coma ‘,’

1. The company wants to raise the salaries of the employees whose salary is below 30000 by giving a bonus of 1000

To do this we need to search the employee entries that have a salary less than 30000 and then add the bonus amount to the salary and update the new value in the data dictionary.

**In this program:**

* First the program reads all the employee details from the given data file and stores in all the data structures respectively by using the method “readDataFileAndProcess”
* The main objective of the program is to compare the CPU efficiency and memory efficiency (Operations under comparison are: Search and Update) we need to start counting the CPU time and initial free memory for Search and Update operations on all the data structures.
* Then to update the salary details and add the bonus amount, it needs to first search the employee records those who have their salaries less than 30000 on each of these data structures by using the methods performListIndexUpdate, performMapUpdate, performCollectionUpdate
* Then add the bonus amount 1000 to their salary
* Then update the new salary details in the data structures.
* End the counter of the CPU time, record the end time of the CPU system clock and again record the free memory at the end
* To calculate the total CPU time taken end time – start time recorded and print the total time.

Similarly, to calculate the memory taken by each of these data structures to perform search and update file is free memory later – free memory before

* **Develop a solution to demonstrate the selected data structures(s) to be most effective.**

In order to find the most effective data structure among

1. Vector
2. Array List
3. Hash Table
4. Hash Map
5. Linked Hash Map
6. Tree Map
7. Hash Set
8. Linked Hash Set
9. Tree Set

We divided the data structures into three categories and calculated their performance individually.

1. List
2. Set
3. Map

In terms of CPU-efficient, When compared individually

**Array list** is efficient among **lists** Vector and Arraylist

**Hash Set** is efficient among **Sets** Hash Set, Linked Hash Set, Tree Set

**Hash Map** is efficient among **Maps** Hash Table, Hash Map, Linked Hash Map, Tree Map

HashMap data structure yields a better CPU efficiency in a search concerned with. HashMap is a collection type which is a key value founded data structure when compared to ArrayList which is a linear type of data structure. Every key will have a value. It gears Map interface of the groups framework.  It contains unique key elements.  It is Unsynchronized, because of this the system doesn’t have to wait for the previous threads to be completed, which makes it fast. This is an benefit when dealing with large data set

In terms of memory-efficient, when compared individually

**Vector** is efficient among lists Vector and Arraylist

**Hash Set** is efficient among Sets Hash Set, Linked Hash Set, Tree Set

**Hash Map** is efficient among Maps Hash Table, Hash Map, Linked Hash Map, Tree Map

Attached the program file and data files on moodle.

**We have 200000 records in the data file it may take long time to get output. Due to this we are attaching screen shot of the output.**

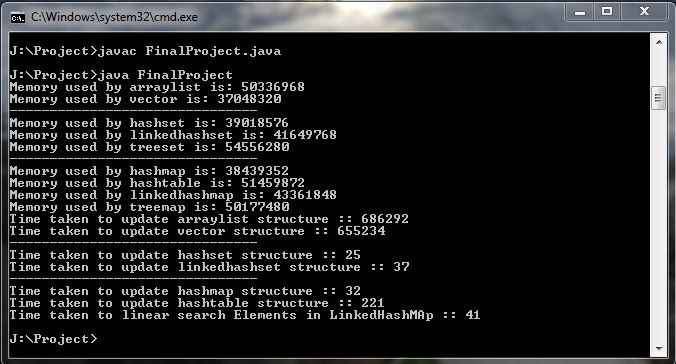
**2. Conclusion**

As per results observed by us based on the requirement we have to choose the data structure. We cannot select a particular data structure that is efficient in CPU efficiency and memory efficiency, as one is good in memory efficiency and one is good in CPU efficient.

If we use list for CPU efficiency array list is the best. If we use sets Hash Set is the best for CPU efficiency. If we use Maps Hash Map is best for CPU efficiency.

If we use list for memory efficiency Vector is the best. If we use sets Hash Set is the best for memory efficiency. If we use Maps Hash table is best for memory efficiency.

**Output:**

****