CS 6301 – Implementation of Advanced Data Structures and Algorithms
- by Dr. Balaji Raghavachari

by Bala Chandra Yadav (bxy140430)
Ravindhar Reddy Thallapureddy (rxt140930)
Mohammad Rafi Shaik (mxs146030)

Implementing multi-dimensional search:

The project involves the implementation of multi-dimensional search. Analyzing the performance of the implementations on varied data set (in size) was the focus.

Problem Statement

Compare the running times of the implementation on various data sets.

Software Requirements (Used)

Java version: Java v1.8.0_31
JDK: Oracle Java SE Development Kit 8
Editor: Eclipse

Instructions to run the code (for detailed, please check readme.txt)

JVM arguments:
-Xms1024M –Xmx1024M
To compile the file
```
>javac LP4Driver.java
```
To run the program use the following command
```
>java LPDriver
```

Input File	AVG RT*(in msec)	Memory
lp4-1.txt	20	17MB/1037MB
lp4-2.txt	14	17MB/1037MB
lp4-3-1k.txt	61	22MB/1037MB
lp4-4-5k.txt	140	40MB/1037MB
lp4-5-ck.txt	5416	229MB/1037MB
lp4-bad.txt	35150	821MB/1037MB

Design choices:

SameSame():

Goal: Finding the number of items whose description contains at least 8 values and its description matches with other item

Assumption: We assumed that samesame() may be called frequently. So, we decided to store the required information in a map for quick retrieval.

Implementation: We have maintained a map with the structure:

Map<Count,Map<Sum, Set<items>>

Count: Number of description values, sum: sum of description values

And this map is updated when

- When a new item is inserted with a description array of size at least 8.
- When an item is updated with a description array of size at least 8.
- When an item having description of size at least 8 is deleted from the storage.