**INTRODUCTION**

Multi-dimensional search:

Consider the web site of a seller like Amazon. They carry tens of thousands of products, and each product has many attributes (Name, Size, Description, Keywords, Manufacturer, Price, etc.). The search engine allows users to specify attributes of products that they are seeking, and shows products that have most of those attributes. To make search efficient, the data is organized using appropriate data structures, such as balanced trees. But, if products are organized by Name, how can search by price implemented efficiently? The solution, called indexing in databases, is to create a new set of references to the objects for each search field, and organize them to implement search operations on that field efficiently. As the objects change, these access structures have to be kept consistent.

Implemented the following functionalities:

1. Insert(id,price,name)
2. Find(id)
3. Delete(id)
4. FindMinPrice(n)
5. FindMaxPrice(n)
6. FindPriceRange(n,low,high)
7. PriceHike(l,h,r)

**SUMMARY**

* Given problem is solved using the following data structures
  + Tree map which is maintained based on the product id’s.
  + Hash table with names as keys and trees as values
  + The values in hashtables are tree maps which are based on the price values.
* The project adheres to the input & output specifications provided in the requirements.

**TEST RESULTS**

java Uxr130130\_P6 < ../../p6-in1.txt

1450.096

java Uxr130130\_P6 < ../../p6-in2.txt

4146.32

java Uxr130130\_P6 < ../../p6-in3-1k.txt

52255.71999999998

java Uxr130130\_P6 < ../../p6-in4-5k.txt

490445.6577750439

java Uxr130130\_P6 < ../../p6-in5-ck.txt

173874678109.581