LP4: Multi-dimensional search

G97

Team Members: Rahul Aravind Mehalingam, Ramesh Suthan Palani, Anandan Sundar, Sanjana Ramakrishnan

In these project we have implemented the Multi-dimensional search using TreeMap and HashMap.

Indexes:
We have built three index using TreeMap to support the queries;

```
public TreeMap<Long, Product> idIndex;
public HashMap<Long, TreeMap<Double, Integer>> descIndex;
public TreeMap<Double, Integer> priceIndex;
```

ID Index:

```
TreeMap<Long, Product> idIndex
Key -> ID
Value ->Product
```

Used for the following operation:

1. Insert
2. Find
3. Delete
4. PriceHike

Description Index:

```
HashMap<Long, TreeMap<Double, Integer>> descIndex

Key -> Description ID
Value -> TreeMap(Key- Price ,Value – count)
```

Used for the following operation:

1. FindMinPrice
2. FindMaxPrice
3. FindPriceRange

Price Index:

```
	public TreeMap<Double, Integer> priceIndex;
Key -> Price
Value ->count
```

Used for the following operation:

1. Range

SameSame Operation:

For SameSame Operation we have used HashMap to find the duplicates and increment the counter.

```
HashMap<Product, Integer> sameSameDescIndex = new HashMap<Product,
Integer>();
```

hashCode() – XOR all the descriptors to generate the hashCode value.

Equals() – Instead of sorting the values in the descriptors and do a pairwise comparison to determine the equality. We have used Hash Set to insert the first objects descriptors. Then for each description id in the second object check whether it is contained in the Hash Set. If any of the descriptor not present in the set then it means two objects are not equal. Else, two objects are equal.