**Data structures and Algorithms**

**E-commerce Platform Search Function**

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

Code :

import java.util.\* ;

import java.util.Collections ;

class Product{

int id ;

String prod\_Name ;

String category ;

int cost ;

Product(){}

Product(int id,String name,String cat,int cost){

this.id=id;

this.prod\_Name=name;

this.category=cat;

this.cost=cost;

}

}

class linearSearch{

public static long LinearSearch(List<Product> prod,String category) {

boolean found=false ;

long start = System.nanoTime();

for(Product lst:prod) {

if(lst.category.equalsIgnoreCase(category)) {

System.out.println("Product ID: " + lst.id + ", Name: " + lst.prod\_Name + ", Category: " + lst.category + ", Cost: " + lst.cost);

found=true;

}

}

long end = System.nanoTime();

if(found==false) System.out.println("Category not found") ;

return end-start ;

}

}

class binarySearch{

public static long BinarySearch(String type,List<Product> prod,String key) {

boolean found=false ;

long start = System.nanoTime();

if(type.equalsIgnoreCase("id"))

{

int idkey=Integer.parseInt(key) ;

int low=0 , high=prod.size()-1 ;

Collections.sort(prod,Comparator.comparingInt(p->p.id)) ;

while(low<=high){

int mid=(low+high)/2 ;

Product midProduct = prod.get(mid);

if(midProduct.id==idkey){

System.out.println("Product ID: " + midProduct.id + ", Name: " + midProduct.prod\_Name + ", Category: " + midProduct.category + ", Cost: " + midProduct.cost);

found=true;

break ;

}

else if(midProduct.id<idkey) low=mid+1 ;

else high=mid-1 ;

}

}

else if(type.equalsIgnoreCase("name")){

Collections.sort(prod,Comparator.comparing(p->p.prod\_Name.toLowerCase())) ;

int low=0 , high=prod.size()-1 ;

while(low<=high){

int mid=(low+high)/2 ;

Product midProduct = prod.get(mid);

int cmp=midProduct.prod\_Name.compareToIgnoreCase(key);

if(cmp==0){

System.out.println("Product ID: " + midProduct.id + ", Name: " + midProduct.prod\_Name + ", Category: " + midProduct.category + ", Cost: " + midProduct.cost);

found=true;

break ;

}

else if(cmp<0) low=mid+1 ;

else high=mid-1 ;

}

}

long end = System.nanoTime();

if(found==false)

System.out.println("not found") ;

return end-start ;

}

}

public class E\_commerce {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

List<Product> prod=new ArrayList<>() ;

prod.add(new Product(1,"Mobile","Gadgets",10000));

prod.add(new Product(2,"TV","Electronics",20000));

prod.add(new Product(3,"Jean","Fashion",1000));

prod.add(new Product(4,"T-shirt","Fashion",200));

prod.add(new Product(5,"Saree","Fashion",700));

prod.add(new Product(6,"Fan","Electronics",20000));

prod.add(new Product(7,"Bluetooth","Gadgets",1200));

long t1=0,t2=0;

int c=0;

while(c<2){

String input=scan.nextLine();

switch(input) {

case "id" :

String idd=scan.nextLine() ;

t1=binarySearch.BinarySearch("id",prod,idd);

break;

case "name":

String namee=scan.nextLine() ;

t1=binarySearch.BinarySearch("name",prod,namee);

break;

case "category":

String cat=scan.nextLine() ;

t2=linearSearch.LinearSearch(prod,cat);

break;

default:

System.out.println("exit");

}

c++ ;

}

System.out.println("Linear search time:" + t2+"ms");

System.out.println("Binary search time:" + t1+"ms");

System.out.println("Binary search takes less time that linear search to search a product");

}

}

