# GTU Department of Computer Engineering

# CSE 222/505 – Spring 2021

# Homework 3 Report


## ŞEYMA NUR CANBAZ

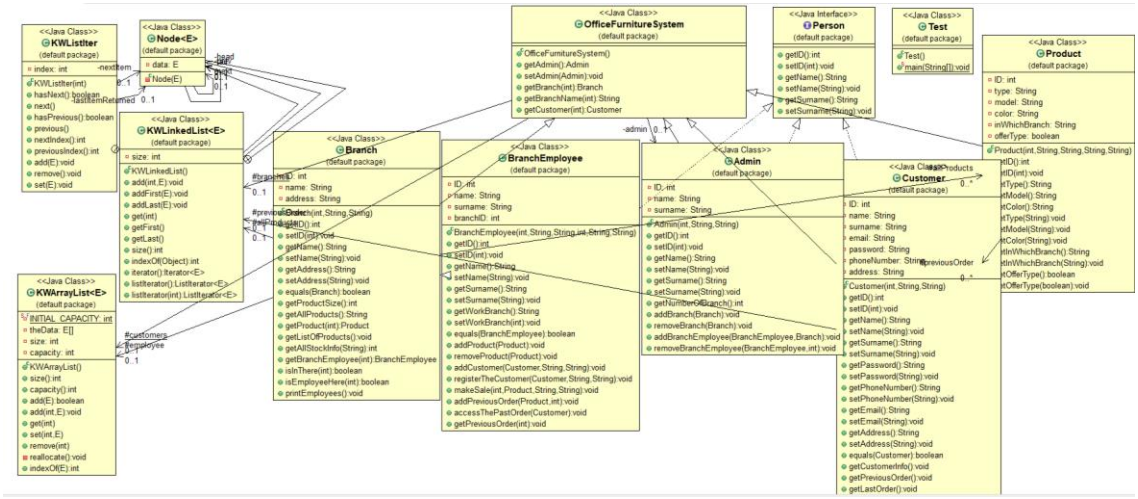
## 171044076

## 1. SYSTEM REQUIREMENTS:

In this assignment, we design an office furniture system. There is a manager managing the system, branch employees and customers using the system. Manager branches can add and remove employees. It can also query the stock of required products. Customers can view the list of products, search for products in the company, see which store the products are available in, shop online or in the store, and view their past orders. Customers must be a member of the system to be able to do all this. When they buy a product, they are asked for their address and telephone information. Branch employees can add and remove products, make sales in the store, access and update the customer's past orders, and create a record if the person who purchases a product from the store is not registered in the system.

For all this, the company needs a class. This class contains information about branches and customers. The manager manages these branches. Therefore, it should have an administrator class. In this class, there are some methods that the manager can add and remove his / her own information, branch and employee. Each branch has different products and employees. Therefore, there must be a branch class to keep this information. The branch employee queries the products in stock, adds and removes products to the system, creates customer orders and makes sales in the store. In order to keep both this information and the branch employee's own information, there must be a branch employee class derived from the branch class. To keep customers' own information and store their orders, it must be customer class. Therefore, there are many interconnected class structures.

## 2. CLASS DIAGRAMS:



## 3. PROBLEM SOLUTION APPROACH:

First, I created an "OfficeFurnitureSystem" class. Then I put the client array, the branch array, and a manager in this class. I needed branch class, manager class, and customer class for this data. First I created the Admin class. I have kept the ID, name and surname of the administrator for this class. I added functions to add and remove branches to the branch array in the "OfficeFurnitureSystem" class. I added methods for adding and removing employees to the BranchEmployee array in the branch class. Then I created the Branch class. I kept the branch ID, branch name, and address in this class. I created an array called allProducts to hold products within the branch. I wrote methods to query the product stocks at the branch. I have created a class named Product to keep product information. Here I kept the information about the product ID, type, model, color, which store it is located in, and if the sale was made, from where the sale was made. I also created a class called BranchEmployee to keep the BranchEmployee array inside the Branch class. In this class, I kept information such as the employee ID, name, surname, and the branch where he works. I have written methods so that the employee can query the product stocks, add and remove products, and make sales. At the same time, I created the addCustomer method to save the customer who bought the product and their past orders into the system. I created the Customer class to keep customers in the system. In this class, I created the previousOrder array to keep customers' past order information. I have added the necessary methods so that the customer can view the product information and place an order. I keep the branch data as KWLinkedList, customers as KWArrayList and products as KWLinkedList.

## 4. TEST CASES:

Test Case 1: Adding the admin in the company.
Test Case 2: Adding the branch in the company.
Test Case 3: Removing branch in the company.
Test Case 4: Adding the same branch in the company.
Test Case 5: Removing a branch that is not in the branch.
Test Case 6: Adding the branch employee in the branch.
Test Case 7: Removing the branch employee in the branch.
Test Case 8: Adding the same branch employee in the same branch again.
Test Case 9: Removing a branch employee that is not in the branch.

Test Case 10: Questioning the product stock with manager.

Test Case 11: Adding a new product in the branch.
Test Case 12: Removing the product in the branch.
Test Case 13: Adding the same product (have same ID) in the company.
Test Case 14: Removing a product that is not in the branch.

Test Case 15: Adding a customer in the company.

Test Case 16: Adding the same customer in the company.

Test Case 17: Seeing the list of product by customer.
Test Case 18: Search the product by customer.
Test Case 19: Search the product that is not exist by customer.
Test Case 20: View the customers previous order (by customer).
Test Case 21: View the customers previous order (by branch employee).
Test Case 22: Shopping online.
Test Case 23: Shopping in branch (with registered customer).
Test Case 24: Shopping in branch (not registered user).
Test Case 25: Updating customers' previous orders.
Test Case 26: Check the stock (in the branch where it is located)

Test Case 27: View the previous order by customer (if he/she has no orders).

Test Case 28: Check the product stock (using product id).

## 5. RUNNING COMMAND AND RESULTS:

Test Case 1: Adding the admin in the company. (Şeyma Canbaz 100000)

```
OfficeFurnitureSystem company = new OfficeFurnitureSystem();
Admin admin = new Admin(100000, "Şeyma", "CANBAZ");
company.setAdmin(admin);
System.out.println("Admin: " + company.getAdmin().getName() + " " + company.getAdmin().getSurname());
```

```
Admin: Şeyma CANBAZ
```

Test Case 2: Adding the branch in the company.

```
Branch b1 = new Branch(2000, "İstanbul", "İstanbul Street");
Branch b2 = new Branch(2001, "Ankara", "Ankara Street");
Branch b3 = new Branch(2002, "İzmir", "İzmir Street");
Branch b4 = new Branch(2003, "Antalya", "Antalya Street");

company.getAdmin().addBranch(b1);
System.out.println("Branch Name: " + company.getAdmin().getBranch(2000).getID() + "\n");

company.getAdmin().addBranch(b2);
System.out.println("Branch Name: " + company.getAdmin().getBranch(2001).getID() + "\n");

company.getAdmin().addBranch(b3);
System.out.println("Branch Name: " + company.getAdmin().getBranch(2002).getID() + "\n");

company.getAdmin().addBranch(b4);
System.out.println("Branch Name: " + company.getAdmin().getBranch(2003).getID() + "\n");
```

```
İstanbul is added
Branch ID: 2000

Ankara is added
Branch ID: 2001

İzmir is added
Branch ID: 2002

Antalya is added
Branch ID: 2003
```

Test Case 3: Removing branch in the company. (İzmir branch)

```
//REMOVE BRANCH
company.getAdmin().removeBrach(b3);
```

```
İzmir branch is removed
```

Test Case 4: Adding the same branch in the company. Throws exception. (İstanbul branch)

```
try {
    //ADD SAME BRANCH
    company.getAdmin().addBranch(b1);   //throws exception
}
catch(Exception e){
    System.out.println(e);
}
```

```
java.util.NoSuchElementException: İstanbul branch is already added!
```

Test Case 5: Removing a branch that is not in the branch. Throws exception. (İzmir branch)

```java
try {
    //REMOVE BRANCH (DOES NOT EXIST)
    company.getAdmin().removeBrach(b3); //throws exception
}
catch(Exception e){
    System.out.println(e);
}
```

```
java.util.NoSuchElementException: There is no branch called İzmir
```

## Test Case 6: Adding the branch employee in the branch.

```java
BranchEmployee BE = new BranchEmployee(600000, "Ali", "CANBAZ", b1.getID(), b1.getName(), b1.getAddress());
BranchEmployee BE2 = new BranchEmployee(600001, "Gülay", "CAN", b1.getID(), b1.getName(), b1.getAddress());

//ADD BRANCH EMPLOYEE
company.getAdmin().addBranchEmployee(BE, company.getBranch(2000).getID());
System.out.println("Branch ID: " + company.getBranch(2000).getBranchEmployee(600000).getWorkBranch());

company.getAdmin().addBranchEmployee(BE2, company.getBranch(2000).getID());
System.out.println("Branch ID: " + company.getBranch(2000).getBranchEmployee(600001).getWorkBranch());
```

```
Ali CANBAZ employee added in this branch
Branch ID: 2000
Gülay CAN employee added in this branch
Branch ID: 2000
```

## Test Case 7: Removing the branch employee in the branch.

```java
//REMOVE BRANCH EMPLOYEE
company.getAdmin().removeBranchEmployee(BE, company.getBranch(2000).getID());
```

```
Ali CANBAZ employee is removed in this branch.
```

## Test Case 8: Adding the same branch employee in the same branch again.

```java
//ADD SAME EMPLOYEE IN THE SAME BRANCH
try {
company.getAdmin().addBranchEmployee(BE2, company.getBranch(2000).getID());
}
catch(Exception e) {
    System.out.println(e);
}
```

```
java.util.NoSuchElementException: Gülay CAN employee is already added in this branch!
```

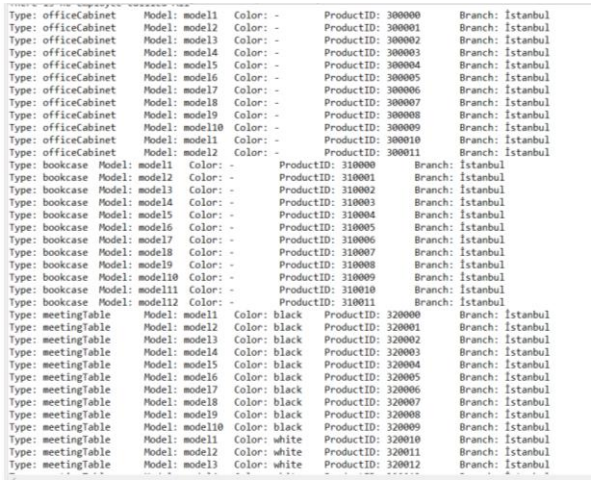
## Test Case 9: Removing a branch employee that is not in the branch.

```java
//REMOVE BRANCH EMPLOYEE (DOES NOT EXIST)
company.getAdmin().removeBranchEmployee(BE, company.getBranch(2000).getID());
```

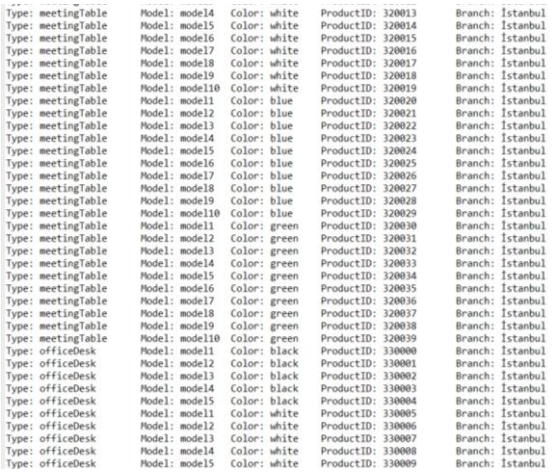
```
There is no employee callled Ali
```

## Test Case 10: Questioning the product stock with admin. (in given branch ID)

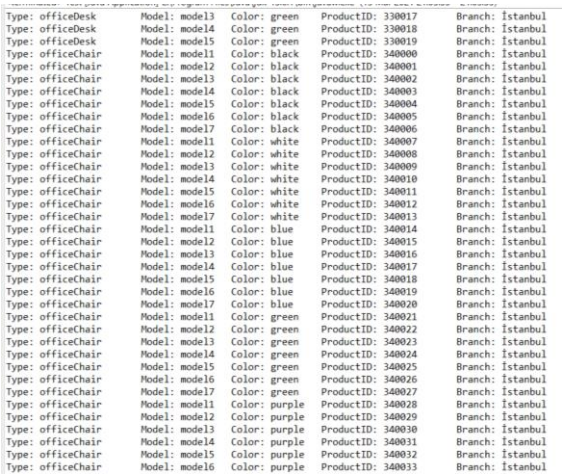
```java
//INQUIRE THE STOCK (BY ADMIN)
company.getAdmin().getBranch(2000).getListofProducts();
```



```
Type: officeCabinet     Model: model1   Color: -        ProductID: 300000       Branch: Istanbul
Type: officeCabinet     Model: model2   Color: -        ProductID: 300001       Branch: Istanbul
Type: officeCabinet     Model: model3   Color: -        ProductID: 300002       Branch: Istanbul
Type: officeCabinet     Model: model4   Color: -        ProductID: 300003       Branch: Istanbul
Type: officeCabinet     Model: model5   Color: -        ProductID: 300004       Branch: Istanbul
Type: officeCabinet     Model: model6   Color: -        ProductID: 300005       Branch: Istanbul
Type: officeCabinet     Model: model7   Color: -        ProductID: 300006       Branch: Istanbul
Type: officeCabinet     Model: model8   Color: -        ProductID: 300007       Branch: Istanbul
Type: officeCabinet     Model: model9   Color: -        ProductID: 300008       Branch: Istanbul
Type: officeCabinet     Model: model10  Color: -        ProductID: 300009       Branch: Istanbul
Type: officeCabinet     Model: model1   Color: -        ProductID: 300010       Branch: Istanbul
Type: officeCabinet     Model: model2   Color: -        ProductID: 300011       Branch: Istanbul
Type: bookcase  Model: model1   Color: -        ProductID: 310000       Branch: Istanbul
Type: bookcase  Model: model2   Color: -        ProductID: 310001       Branch: Istanbul
Type: bookcase  Model: model3   Color: -        ProductID: 310002       Branch: Istanbul
Type: bookcase  Model: model4   Color: -        ProductID: 310003       Branch: Istanbul
Type: bookcase  Model: model5   Color: -        ProductID: 310004       Branch: Istanbul
Type: bookcase  Model: model6   Color: -        ProductID: 310005       Branch: Istanbul
Type: bookcase  Model: model7   Color: -        ProductID: 310006       Branch: Istanbul
Type: bookcase  Model: model8   Color: -        ProductID: 310007       Branch: Istanbul
Type: bookcase  Model: model9   Color: -        ProductID: 310008       Branch: Istanbul
Type: bookcase  Model: model10  Color: -        ProductID: 310009       Branch: Istanbul
Type: bookcase  Model: model11  Color: -        ProductID: 310010       Branch: Istanbul
Type: bookcase  Model: model12  Color: -        ProductID: 310011       Branch: Istanbul
Type: meetingTable      Model: model1   Color: black    ProductID: 320000       Branch: Istanbul
Type: meetingTable      Model: model2   Color: black    ProductID: 320001       Branch: Istanbul
Type: meetingTable      Model: model3   Color: black    ProductID: 320002       Branch: Istanbul
Type: meetingTable      Model: model4   Color: black    ProductID: 320003       Branch: Istanbul
Type: meetingTable      Model: model5   Color: black    ProductID: 320004       Branch: Istanbul
Type: meetingTable      Model: model6   Color: black    ProductID: 320005       Branch: Istanbul
Type: meetingTable      Model: model7   Color: black    ProductID: 320006       Branch: Istanbul
Type: meetingTable      Model: model8   Color: black    ProductID: 320007       Branch: Istanbul
Type: meetingTable      Model: model9   Color: black    ProductID: 320008       Branch: Istanbul
Type: meetingTable      Model: model10  Color: black    ProductID: 320009       Branch: Istanbul
Type: meetingTable      Model: model1   Color: white    ProductID: 320010       Branch: Istanbul
Type: meetingTable      Model: model2   Color: white    ProductID: 320011       Branch: Istanbul
Type: meetingTable      Model: model3   Color: white    ProductID: 320012       Branch: Istanbul
```



```
Type: meetingTable      Model: model4   Color: white    ProductID: 320013       Branch: Istanbul
Type: meetingTable      Model: model5   Color: white    ProductID: 320014       Branch: Istanbul
Type: meetingTable      Model: model6   Color: white    ProductID: 320015       Branch: Istanbul
Type: meetingTable      Model: model7   Color: white    ProductID: 320016       Branch: Istanbul
Type: meetingTable      Model: model8   Color: white    ProductID: 320017       Branch: Istanbul
Type: meetingTable      Model: model9   Color: white    ProductID: 320018       Branch: Istanbul
Type: meetingTable      Model: model10  Color: white    ProductID: 320019       Branch: Istanbul
Type: meetingTable      Model: model1   Color: blue     ProductID: 320020       Branch: Istanbul
Type: meetingTable      Model: model2   Color: blue     ProductID: 320021       Branch: Istanbul
Type: meetingTable      Model: model3   Color: blue     ProductID: 320022       Branch: Istanbul
Type: meetingTable      Model: model4   Color: blue     ProductID: 320023       Branch: Istanbul
Type: meetingTable      Model: model5   Color: blue     ProductID: 320024       Branch: Istanbul
Type: meetingTable      Model: model6   Color: blue     ProductID: 320025       Branch: Istanbul
Type: meetingTable      Model: model7   Color: blue     ProductID: 320026       Branch: Istanbul
Type: meetingTable      Model: model8   Color: blue     ProductID: 320027       Branch: Istanbul
Type: meetingTable      Model: model9   Color: blue     ProductID: 320028       Branch: Istanbul
Type: meetingTable      Model: model10  Color: blue     ProductID: 320029       Branch: Istanbul
Type: meetingTable      Model: model1   Color: green    ProductID: 320030       Branch: Istanbul
Type: meetingTable      Model: model2   Color: green    ProductID: 320031       Branch: Istanbul
Type: meetingTable      Model: model3   Color: green    ProductID: 320032       Branch: Istanbul
Type: meetingTable      Model: model4   Color: green    ProductID: 320033       Branch: Istanbul
Type: meetingTable      Model: model5   Color: green    ProductID: 320034       Branch: Istanbul
Type: meetingTable      Model: model6   Color: green    ProductID: 320035       Branch: Istanbul
Type: meetingTable      Model: model7   Color: green    ProductID: 320036       Branch: Istanbul
Type: meetingTable      Model: model8   Color: green    ProductID: 320037       Branch: Istanbul
Type: meetingTable      Model: model9   Color: green    ProductID: 320038       Branch: Istanbul
Type: meetingTable      Model: model10  Color: green    ProductID: 320039       Branch: Istanbul
Type: officeDesk        Model: model1   Color: black    ProductID: 330000       Branch: Istanbul
Type: officeDesk        Model: model2   Color: black    ProductID: 330001       Branch: Istanbul
Type: officeDesk        Model: model3   Color: black    ProductID: 330002       Branch: Istanbul
Type: officeDesk        Model: model4   Color: black    ProductID: 330003       Branch: Istanbul
Type: officeDesk        Model: model5   Color: black    ProductID: 330004       Branch: Istanbul
Type: officeDesk        Model: model1   Color: white    ProductID: 330005       Branch: Istanbul
Type: officeDesk        Model: model2   Color: white    ProductID: 330006       Branch: Istanbul
Type: officeDesk        Model: model3   Color: white    ProductID: 330007       Branch: Istanbul
Type: officeDesk        Model: model4   Color: white    ProductID: 330008       Branch: Istanbul
Type: officeDesk        Model: model5   Color: white    ProductID: 330009       Branch: Istanbul
```



```
Type: officeDesk        Model: model3   Color: green    ProductID: 330017       Branch: Istanbul
Type: officeDesk        Model: model4   Color: green    ProductID: 330018       Branch: Istanbul
Type: officeDesk        Model: model5   Color: green    ProductID: 330019       Branch: Istanbul
Type: officeChair       Model: model1   Color: black    ProductID: 340000       Branch: Istanbul
Type: officeChair       Model: model2   Color: black    ProductID: 340001       Branch: Istanbul
Type: officeChair       Model: model3   Color: black    ProductID: 340002       Branch: Istanbul
Type: officeChair       Model: model4   Color: black    ProductID: 340003       Branch: Istanbul
Type: officeChair       Model: model5   Color: black    ProductID: 340004       Branch: Istanbul
Type: officeChair       Model: model6   Color: black    ProductID: 340005       Branch: Istanbul
Type: officeChair       Model: model7   Color: black    ProductID: 340006       Branch: Istanbul
Type: officeChair       Model: model1   Color: white    ProductID: 340007       Branch: Istanbul
Type: officeChair       Model: model2   Color: white    ProductID: 340008       Branch: Istanbul
Type: officeChair       Model: model3   Color: white    ProductID: 340009       Branch: Istanbul
Type: officeChair       Model: model4   Color: white    ProductID: 340010       Branch: Istanbul
Type: officeChair       Model: model5   Color: white    ProductID: 340011       Branch: Istanbul
Type: officeChair       Model: model6   Color: white    ProductID: 340012       Branch: Istanbul
Type: officeChair       Model: model7   Color: white    ProductID: 340013       Branch: Istanbul
Type: officeChair       Model: model1   Color: blue     ProductID: 340014       Branch: Istanbul
Type: officeChair       Model: model2   Color: blue     ProductID: 340015       Branch: Istanbul
Type: officeChair       Model: model3   Color: blue     ProductID: 340016       Branch: Istanbul
Type: officeChair       Model: model4   Color: blue     ProductID: 340017       Branch: Istanbul
Type: officeChair       Model: model5   Color: blue     ProductID: 340018       Branch: Istanbul
Type: officeChair       Model: model6   Color: blue     ProductID: 340019       Branch: Istanbul
Type: officeChair       Model: model7   Color: blue     ProductID: 340020       Branch: Istanbul
Type: officeChair       Model: model1   Color: green    ProductID: 340021       Branch: Istanbul
Type: officeChair       Model: model2   Color: green    ProductID: 340022       Branch: Istanbul
Type: officeChair       Model: model3   Color: green    ProductID: 340023       Branch: Istanbul
Type: officeChair       Model: model4   Color: green    ProductID: 340024       Branch: Istanbul
Type: officeChair       Model: model5   Color: green    ProductID: 340025       Branch: Istanbul
Type: officeChair       Model: model6   Color: green    ProductID: 340026       Branch: Istanbul
Type: officeChair       Model: model7   Color: green    ProductID: 340027       Branch: Istanbul
Type: officeChair       Model: model1   Color: purple   ProductID: 340028       Branch: Istanbul
Type: officeChair       Model: model2   Color: purple   ProductID: 340029       Branch: Istanbul
Type: officeChair       Model: model3   Color: purple   ProductID: 340030       Branch: Istanbul
Type: officeChair       Model: model4   Color: purple   ProductID: 340031       Branch: Istanbul
Type: officeChair       Model: model5   Color: purple   ProductID: 340032       Branch: Istanbul
Type: officeChair       Model: model6   Color: purple   ProductID: 340033       Branch: Istanbul
```

## Test Case 11: Adding a new product in the branch.

```
Product product1 = new Product(5555, "bookcase", "model2", "-", company.getAdmin().getBranch(2000).getID());
Product product2 = new Product(5565, "bookcase", "model2", "-", company.getAdmin().getBranch(2000).getID());
Product product3 = new Product(5575, "bookcase", "model2", "-", company.getAdmin().getBranch(2000).getID());
Product product4 = new Product(5275, "meetingTable", "model2", "-", company.getAdmin().getBranch(2000).getID());
Product product5 = new Product(5175, "officeChair", "model2", "-", company.getAdmin().getBranch(2000).getID());
Product product6 = new Product(6175, "officeChair", "model2", "-", company.getAdmin().getBranch(2000).getID());

//ADD PRODUCT
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addProduct(product1, 2000);
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addProduct(product2, 2000);
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addProduct(product3, 2000);
```

(Show the product ID)

```
5555product is added!
5565product is added!
5575product is added!
```

Test Case 12: Removing the product in the branch.

```
//REMOVE PRODUCT
company.getAdmin().getBranch(2000).getBranchEmployee(600001).removeProduct(product1, 2000);
```

```
5555 product is removed
```

Test Case 13: Adding the same product (have same ID) in the company.

```
//ADD PRODUCT(SAME ID)
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addProduct(product3, 2000);
```

```
5575product is already added. Please change the product ID
```

Test Case 14: Removing a product that is not in the branch. (controls the product ID)

```
//REMOVE PRODUCT (DOES NOT EXIST)
company.getAdmin().getBranch(2000).getBranchEmployee(600001).removeProduct(product1, 2000);
```

```
There is no product called 5555
```

Test Case 15: Adding a customer in the company.

```
//ADD CUSTOMER
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addCustomer(customer1, "cansu@email.com", "12345");
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addCustomer(customer2, "zeynep@email.com", "zeynep");
```

```
Cansu Tan customer is added
Email and password saved! Registration completed successfully
Zeynep Sevim customer is added
Email and password saved! Registration completed successfully
```

Test Case 16: Adding the same customer in the company. (with the same id)

```
//ADD CUSTOMER(SAME CUSTOMER ID)
company.getAdmin().getBranch(2000).getBranchEmployee(600001).addCustomer(customer1, "cansu@email.com", "12345");
```

```
Cansu Tan customer is already added!
```

Test Case 17: Seeing the list of product by customer. (in given branch)

```
//SEE THE LIST OF PRODUCT (CUSTOMER)
company.getCustomer(900001).seeListOfProducts(b1);
```

```
******LIST OF THE PRODUCT*******
Type: officeCabinet     Model: model1   Color: -       ProductID: 300000      Branch: İstanbul
Type: officeCabinet     Model: model2   Color: -       ProductID: 300001      Branch: İstanbul
Type: officeCabinet     Model: model3   Color: -       ProductID: 300002      Branch: İstanbul
Type: officeCabinet     Model: model4   Color: -       ProductID: 300003      Branch: İstanbul
Type: officeCabinet     Model: model5   Color: -       ProductID: 300004      Branch: İstanbul
Type: officeCabinet     Model: model6   Color: -       ProductID: 300005      Branch: İstanbul
Type: officeCabinet     Model: model7   Color: -       ProductID: 300006      Branch: İstanbul
Type: officeCabinet     Model: model8   Color: -       ProductID: 300007      Branch: İstanbul
Type: officeCabinet     Model: model9   Color: -       ProductID: 300008      Branch: İstanbul
Type: officeCabinet     Model: model10  Color: -       ProductID: 300009      Branch: İstanbul
Type: officeCabinet     Model: model11  Color: -       ProductID: 300010      Branch: İstanbul
Type: officeCabinet     Model: model2   Color: -       ProductID: 300011      Branch: İstanbul
Type: bookcase  Model: model1   Color: -       ProductID: 310000      Branch: İstanbul
Type: bookcase  Model: model2   Color: -       ProductID: 310001      Branch: İstanbul
Type: bookcase  Model: model3   Color: -       ProductID: 310002      Branch: İstanbul
Type: bookcase  Model: model4   Color: -       ProductID: 310003      Branch: İstanbul
Type: bookcase  Model: model5   Color: -       ProductID: 310004      Branch: İstanbul
Type: bookcase  Model: model6   Color: -       ProductID: 310005      Branch: İstanbul
Type: bookcase  Model: model7   Color: -       ProductID: 310006      Branch: İstanbul
Type: bookcase  Model: model8   Color: -       ProductID: 310007      Branch: İstanbul
Type: bookcase  Model: model9   Color: -       ProductID: 310008      Branch: İstanbul
Type: bookcase  Model: model10  Color: -       ProductID: 310009      Branch: İstanbul
Type: bookcase  Model: model11  Color: -       ProductID: 310010      Branch: İstanbul
Type: bookcase  Model: model12  Color: -       ProductID: 310011      Branch: İstanbul
Type: meetingTable      Model: model1   Color: black    ProductID: 320000      Branch: İstanbul
Type: meetingTable      Model: model2   Color: black    ProductID: 320001      Branch: İstanbul
Type: meetingTable      Model: model3   Color: black    ProductID: 320002      Branch: İstanbul
Type: meetingTable      Model: model4   Color: black    ProductID: 320003      Branch: İstanbul
Type: meetingTable      Model: model5   Color: black    ProductID: 320004      Branch: İstanbul
Type: meetingTable      Model: model6   Color: black    ProductID: 320005      Branch: İstanbul
Type: meetingTable      Model: model7   Color: black    ProductID: 320006      Branch: İstanbul
Type: meetingTable      Model: model8   Color: black    ProductID: 320007      Branch: İstanbul
Type: meetingTable      Model: model9   Color: black    ProductID: 320008      Branch: İstanbul
Type: meetingTable      Model: model10  Color: black    ProductID: 320009      Branch: İstanbul
Type: meetingTable      Model: model1   Color: white    ProductID: 320010      Branch: İstanbul
Type: meetingTable      Model: model2   Color: white    ProductID: 320011      Branch: İstanbul
Type: meetingTable      Model: model5   Color: white    ProductID: 320014      Branch: İstanbul
Type: meetingTable      Model: model6   Color: white    ProductID: 320015      Branch: İstanbul
Type: meetingTable      Model: model7   Color: white    ProductID: 320016      Branch: İstanbul
Type: meetingTable      Model: model8   Color: white    ProductID: 320017      Branch: İstanbul
Type: meetingTable      Model: model9   Color: white    ProductID: 320018      Branch: İstanbul
Type: meetingTable      Model: model10  Color: white    ProductID: 320019      Branch: İstanbul
Type: meetingTable      Model: model1   Color: blue     ProductID: 320020      Branch: İstanbul
Type: meetingTable      Model: model2   Color: blue     ProductID: 320021      Branch: İstanbul
Type: meetingTable      Model: model3   Color: blue     ProductID: 320022      Branch: İstanbul
Type: meetingTable      Model: model4   Color: blue     ProductID: 320023      Branch: İstanbul
Type: meetingTable      Model: model5   Color: blue     ProductID: 320024      Branch: İstanbul
Type: meetingTable      Model: model6   Color: blue     ProductID: 320025      Branch: İstanbul
Type: meetingTable      Model: model7   Color: blue     ProductID: 320026      Branch: İstanbul
Type: meetingTable      Model: model8   Color: blue     ProductID: 320027      Branch: İstanbul
Type: meetingTable      Model: model9   Color: blue     ProductID: 320028      Branch: İstanbul
Type: meetingTable      Model: model10  Color: blue     ProductID: 320029      Branch: İstanbul
Type: meetingTable      Model: model1   Color: green    ProductID: 320030      Branch: İstanbul
Type: meetingTable      Model: model2   Color: green    ProductID: 320031      Branch: İstanbul
Type: meetingTable      Model: model3   Color: green    ProductID: 320032      Branch: İstanbul
Type: meetingTable      Model: model4   Color: green    ProductID: 320033      Branch: İstanbul
Type: meetingTable      Model: model5   Color: green    ProductID: 320034      Branch: İstanbul
Type: meetingTable      Model: model6   Color: green    ProductID: 320035      Branch: İstanbul
Type: meetingTable      Model: model7   Color: green    ProductID: 320036      Branch: İstanbul
Type: meetingTable      Model: model8   Color: green    ProductID: 320037      Branch: İstanbul
Type: meetingTable      Model: model9   Color: green    ProductID: 320038      Branch: İstanbul
Type: meetingTable      Model: model10  Color: green    ProductID: 320039      Branch: İstanbul
Type: officeDesk        Model: model1   Color: black    ProductID: 330000      Branch: İstanbul
Type: officeDesk        Model: model2   Color: black    ProductID: 330001      Branch: İstanbul
Type: officeDesk        Model: model3   Color: black    ProductID: 330002      Branch: İstanbul
Type: officeDesk        Model: model4   Color: black    ProductID: 330003      Branch: İstanbul
Type: officeDesk        Model: model5   Color: black    ProductID: 330004      Branch: İstanbul
Type: officeDesk        Model: model1   Color: white    ProductID: 330005      Branch: İstanbul
Type: officeDesk        Model: model2   Color: white    ProductID: 330006      Branch: İstanbul
Type: officeDesk        Model: model3   Color: white    ProductID: 330007      Branch: İstanbul
Type: officeDesk        Model: model4   Color: white    ProductID: 330008      Branch: İstanbul
Type: officeDesk        Model: model5   Color: white    ProductID: 330009      Branch: İstanbul
Type: officeDesk        Model: model1   Color: blue     ProductID: 330010      Branch: İstanbul
Type: officeChair       Model: model3   Color: black    ProductID: 340002      Branch: İstanbul
Type: officeChair       Model: model4   Color: black    ProductID: 340003      Branch: İstanbul
Type: officeChair       Model: model5   Color: black    ProductID: 340004      Branch: İstanbul
Type: officeChair       Model: model6   Color: black    ProductID: 340005      Branch: İstanbul
Type: officeChair       Model: model7   Color: black    ProductID: 340006      Branch: İstanbul
Type: officeChair       Model: model1   Color: white    ProductID: 340007      Branch: İstanbul
Type: officeChair       Model: model2   Color: white    ProductID: 340008      Branch: İstanbul
Type: officeChair       Model: model3   Color: white    ProductID: 340009      Branch: İstanbul
Type: officeChair       Model: model4   Color: white    ProductID: 340010      Branch: İstanbul
Type: officeChair       Model: model5   Color: white    ProductID: 340011      Branch: İstanbul
Type: officeChair       Model: model6   Color: white    ProductID: 340012      Branch: İstanbul
Type: officeChair       Model: model7   Color: white    ProductID: 340013      Branch: İstanbul
Type: officeChair       Model: model1   Color: blue     ProductID: 340014      Branch: İstanbul
Type: officeChair       Model: model2   Color: blue     ProductID: 340015      Branch: İstanbul
Type: officeChair       Model: model3   Color: blue     ProductID: 340016      Branch: İstanbul
Type: officeChair       Model: model4   Color: blue     ProductID: 340017      Branch: İstanbul
Type: officeChair       Model: model5   Color: blue     ProductID: 340018      Branch: İstanbul
Type: officeChair       Model: model6   Color: blue     ProductID: 340019      Branch: İstanbul
Type: officeChair       Model: model7   Color: blue     ProductID: 340020      Branch: İstanbul
Type: officeChair       Model: model1   Color: green    ProductID: 340021      Branch: İstanbul
Type: officeChair       Model: model2   Color: green    ProductID: 340022      Branch: İstanbul
Type: officeChair       Model: model3   Color: green    ProductID: 340023      Branch: İstanbul
Type: officeChair       Model: model4   Color: green    ProductID: 340024      Branch: İstanbul
Type: officeChair       Model: model5   Color: green    ProductID: 340025      Branch: İstanbul
Type: officeChair       Model: model6   Color: green    ProductID: 340026      Branch: İstanbul
Type: officeChair       Model: model7   Color: green    ProductID: 340027      Branch: İstanbul
Type: officeChair       Model: model1   Color: purple   ProductID: 340028      Branch: İstanbul
Type: officeChair       Model: model2   Color: purple   ProductID: 340029      Branch: İstanbul
Type: officeChair       Model: model3   Color: purple   ProductID: 340030      Branch: İstanbul
Type: officeChair       Model: model4   Color: purple   ProductID: 340031      Branch: İstanbul
Type: officeChair       Model: model5   Color: purple   ProductID: 340032      Branch: İstanbul
Type: officeChair       Model: model6   Color: purple   ProductID: 340033      Branch: İstanbul
Type: officeChair       Model: model7   Color: purple   ProductID: 340034      Branch: İstanbul
Type: bookcase  Model: model2   Color: -       ProductID: 5565 Branch: İstanbul
Type: bookcase  Model: model2   Color: -       ProductID: 5575 Branch: İstanbul
```

## Test Case 18: Search the product by customer.

```
//SEARCH THE PRODUCT (CUSTOMER)
company.getCustomer(900001).searchProduct(5565);
System.out.println("\n");
```

```
Searching...
Product in İstanbul branch
```

## Test Case 19: Search the product that is not exist by customer.

```
//CUSTOMER SEARCH THE PRODUCT (DOES NOT EXIST)
company.getCustomer(900001).searchProduct(6175);
System.out.println("\n");
```

```
Searching...
Product not found
```

## Test Case 20: Shopping online.

```
//SHOPPING ONLINE
company.getCustomer(900001).shopOnline(product5, 600001, "Ceviz Street NO:15", "777 77 77");
```

```
The product added to the past sales list
5175 product is removed
Your order has been received!

PRODUCT ORDER INFORMATION
Product Order ID: 5175
Product Order Type: officeChair
Product Order Model: model2
Product Order Color: -
Product Order Status: online
Product Order in BranchİstanbuL

CUSTOMER INFORMATION
ID: 900001
NAME: Zeynep
SURNAME: Sevim
ADDRESS: Ceviz Street NO:15
PHONE NUMBER: 777 77 77
```

## Test Case 21: View the previous order (by customer).

```
//ACCESS THE PAST ORDER (BY CUSTOMER)
System.out.println("--YOUR PREVIOUS ORDERS--");
company.getCustomer(900001).getPreviousOrder();
System.out.println();
```

```
--YOUR PREVIOUS ORDERS--
1. PRODUCT
Product ID: 5175 Type: officeChair Model: model2 Color: - Status: online Order in Branch: İstanbul
```

## Test Case 22: View the customers' previous order (by branch employee).

```
//ACCESS THE PAST ORDER (BY CUSTOMER)
System.out.println("--YOUR PREVIOUS ORDERS--");
company.getCustomer(900000).getPreviousOrder();
System.out.println();
```

```
1. PRODUCT
Product ID: 5175 Type: officeChair Model: model2 Color: - Status: online Order in Branch: İstanbul
```

## Test Case 23: Shopping in branch (with registered customer).

```
//SHOP IN BRANCH (WITH REGISTERED CUSTOMER)
company.getAdmin().getBranch(2000).getBranchEmployee(600001).makeSale(900001, product6, "Yıldız Street", "374 47 24");
```

```
The product added to the past sales list
6175 product is removed

CUSTOMER INFORMATION
ID: 900001
NAME: Zeynep
SURNAME: Sevim
ADDRESS: Yıldız Street
PHONE NUMBER: 374 47 24

The product sold
```

## Test Case 24: Shopping in branch (not registered user).

```
//SHOP IN BRANCH (WITH NOT REGISTERED CUSTOMER)
company.getAdmin().getBranch(2000).getBranchEmployee(600001).makeSale(9000, product6, "Yıldız Street", "374 47 24");
System.out.println("\n");
```

```
Customer not found! Please register
```

## Test Case 25: Updating customers' previous orders.

```
public void makeSale(int customerID, Product orderProduct, String customerAddress, String customerPhone) {
    int flag = 0;

    for(Customer cs : customer) {
        if(cs.getID() == customerID) {
            cs.setAddress(customerAddress);
            cs.setPhoneNumber(customerPhone);
            addPreviousOrder(orderProduct, customerID);
            removeProduct(orderProduct, branchID);
            cs.getCustomerInfo();
            flag++;
            System.out.println("The product sold");
        }
    }
    if(flag == 0) {
        System.out.println("Customer not found! Please register");
    }
}
```

```
company.getAdmin().getBranch(2000).getBranchEmployee(600001).accessThePastOrder(customer2);
```

```
1. PRODUCT
Product ID: 5175 Type: officeChair Model: model2 Color: - Status: online Order in Branch: İstanbul
2. PRODUCT
Product ID: 6175 Type: officeChair Model: model2 Color: - Status: online Order in Branch: İstanbul
```

## Test Case 26: Check the stock (in the branch where it is located).

```
//CONTROL THE STOCK
System.out.println("Office cabinet stock: " + company.getAdmin().getBranch(2000).getAllStockInfo("officeCabinet"));
System.out.println("Office chair stock: " + company.getAdmin().getBranch(2000).getAllStockInfo("officeChair"));
System.out.println("Office desk stock: " + company.getAdmin().getBranch(2000).getAllStockInfo("officeDesk"));
System.out.println("Bookcase stock: " + company.getAdmin().getBranch(2000).getAllStockInfo("bookcase"));
System.out.println("Meeting table stock: " + company.getAdmin().getBranch(2000).getAllStockInfo("meetingTable"));
System.out.println("\n");
```

```
Office cabinet stock: 12
Office chair stock: 36
Office desk stock: 20
Bookcase stock: 15
Meeting table stock: 41
```

## Test Case 27: View the previous order by customer (if he/she has no orders).

```
//ACCESS THE PAST ORDER BY CUSTOMER (NO PREVIOUS ORDER)
System.out.println("--YOUR PREVIOUS ORDERS--");
company.getCustomer(900000).getPreviousOrder();
System.out.println();
```

```
--YOUR PREVIOUS ORDERS--
Order not found
```

## Test Case 28: Check the product stock (using product id).

```
There is no product called 5175
There is no product called 6175
```

## TIME COMPLEXITIES:

1. Add Branch:

```
public void addBranch(Branch newBranch){
    if(branches.indexOf(newBranch) != -1)
        System.out.println(newBranch.getName() + " branch is already added!");
    else {
        branches.addLast(newBranch);
        System.out.println(newBranch.getName() + " branch is added");
    }
}
```

T(n) = theta(n)

2. Remove branch:

```
public void removeBranch(Branch removedBranch) throws NoSuchElementException {

    if(branches.indexOf(removedBranch) == -1)
        throw new NoSuchElementException(removedBranch.getName() + " branch not found");
    else {
        branches.remove(removedBranch);
        System.out.println(removedBranch.getName() + " branch is removed");
    }
}
```

T(n) = theta(n)

3. Add branch employee:

```
public void addBranchEmployee(BranchEmployee newEmployee, Branch branch) {
    for(int i=0; i<branches.size(); i++) {
        if(branches.get(i).getID() == branch.getID()) {
            if(branches.get(i).isEmployeeHere(newEmployee.getID())) {
                System.out.println(newEmployee.getName() + " " + newEmployee.getSurname() + " employee is already addded in this branch");
            }
            else {
                branches.get(i).employee.add(newEmployee);
                System.out.println(newEmployee.getName() + " " + newEmployee.getSurname() + " employee is added");
            }
        }
    }
}
```

T(n) = O(n^2 x m)

n = branches size

m = emplyee size

4. Remove branch employee:

```
public void removeBranchEmployee(BranchEmployee removedEmployee, int branchID) throws NoSuchElementException {
    int flag = 0;
    for(int i=0; i<branches.size(); i++) {
        if(branches.get(i).getID() == branchID) {
            if(branches.get(i).isEmployeeHere(removedEmployee.getID())) {
                int index = branches.get(i).employee.indexOf(removedEmployee);
                branches.get(i).employee.remove(index);
                System.out.println(removedEmployee.getName() + " " + removedEmployee.getSurname()
                                   + " employee is removed");
                flag++;
            }
        }
    }
    if(flag == 0)
        throw new NoSuchElementException("Employee is not in this branch!");
}
}
```

T(n) = O(n^3 x m)

n = branches size

m = emplyee size

5. Add product:

```
public void addProduct(Product newProduct) {
    if(allProducts.indexOf(newProduct) != -1)
        System.out.println("Product is already added!");
    else {
        allProducts.addLast(newProduct);
        System.out.println(newProduct.getID() + " product is added");
    }
}
```

T(n) = theta(n)

6. Remove product:

```
public void removeProduct(Product removedProduct) {
    if(allProducts.indexOf(removedProduct) == -1)
        throw new NoSuchElementException(removedProduct.getID() + "product not found");
    else {
        allProducts.remove(removedProduct);
    }
}
```

T(n) = theta(n)

## 7. Add customer:

```
public void addCustomer(Customer newCustomer, String email, String password) {
    //TODO
    if(customers.size() == 0) {
        customers.add(newCustomer);
        System.out.println(newCustomer.getName() + " " + newCustomer.getSurname() + " customer is added");
        registerTheCustomer(newCustomer, email, password);
    }
    else {
        boolean flag = false;
        for(int i=0; i<customers.size(); i++) {
            if(customers.get(i).getID() == newCustomer.getID()) {
                System.out.println(newCustomer.getName() + " " + newCustomer.getSurname() + " is already registered!");
                flag = true;
            }
        }
        if(flag == false) {
            customers.add(newCustomer);
            System.out.println(newCustomer.getName() + " " + newCustomer.getSurname() + " customer is added");
            registerTheCustomer(newCustomer, email, password);
        }
    }
}
```

T(n) = theta(n)

## 8. Search product:

```
public void searchProduct(int productID) {
    int flag = 0;
    System.out.println("Searching...");

    for(int i=0; i<branches.size(); i++) {
        if(branches.get(i).isInThere(productID)) {
            for(int j=0; j<branches.get(i).allProducts.size(); j++) {
                if(branches.get(i).allProducts.get(j).getID() == productID) {
                    System.out.println("Product in " + branches.get(i).getName() + " branch");
                    flag++;
                }
            }
        }
    }

    if(flag == 0)
        System.out.println(productID + " product not found");
}
```

T(n) = O(n^3 x m)

n: number of branches

m: number of products

## 9. See the list of products:

```
public void seeListOfProducts(int inBranch) {
    for(int i=0; i<branches.size(); i++) {
        if(branches.get(i).getID() == inBranch)
            System.out.println(branches.get(i).getAllProducts());
    }
}

public String getAllProducts() {
    String str = "";

    for(int i=0; i<allProducts.size(); i++) {
        str += "Type: " + allProducts.get(i).getType() + "\t"
                + "Model: " + allProducts.get(i).getModel() + "\t"
                + "Color: " + allProducts.get(i).getColor() + "\t"
                + "ProductID: " + allProducts.get(i).getID() + "\t"
                + "Branch: " +  allProducts.get(i).getInWhichBranch() + "\n" ;
    }
    return str;
}
```

T(n) = theta(n)

## 10. Make sales:

```java
public void makeSale(int customerID, Product orderProduct, String customerAddress, String customerPhone) {
    int flag = 0;

    for(int i=0; i<customers.size(); i++) {
        if(customers.get(i).getID() == customerID) {
            customers.get(i).setAddress(customerAddress);
            customers.get(i).setPhoneNumber(customerPhone);
            addPreviousOrder(orderProduct, customerID);
            removeProduct(orderProduct);
            customers.get(i).getCustomerInfo();
            flag++;
            System.out.println("The product sold");
        }
    }
    if(flag == 0) {
        System.out.println("Customer not found! Please register");
    }
}
```

T(n) = O(n)

## 11. Shopping online:

```java
public void shopOnline(Product product, int employeeID, String customerAddress, String customerPhone) {
    for(int i=0; i<branches.size(); i++) {
        if(branches.get(i).isInThere(product.getID())) {
            this.address = customerAddress;
            this.phoneNumber = customerPhone;
            branches.get(i).getBranchEmployee(employeeID).addPreviousOrder(product, this.getID());
            branches.get(i).getBranchEmployee(employeeID).removeProduct(product);
            System.out.println("Your order has been received!");
            this.getLastOrder();
            this.getCustomerInfo();
        }
    }
}
```

T(n) = theta(n^2 x m^3)

n: number of branch

m: number of previous order

## 12. See the previous order:

```java
public void getPreviousOrder() {
    if(previousOrder.size() == 0)
        System.out.println("Orders not found");
    else {
        String status = "";

        for(int i=0; i<previousOrder.size(); i++) {
            if(previousOrder.get(i).getOfferType())
                status = "online";
            else
                status = "inBranch";

            System.out.println(i+1 + ". PRODUCT\n"
                    + "Product ID: " + previousOrder.get(i).getID() + " "
                    + "Type: " + previousOrder.get(i).getType() + " "
                    + "Model: " + previousOrder.get(i).getModel() + " "
                    + "Color: " + previousOrder.get(i).getColor() + " "
                    + "Status: " + status + " "
                    + "Order in Branch: " + previousOrder.get(i).getInWhichBranch());
        }
    }
}
```

T(n) = O(n^2)

## 13. Add previous order:

```java
public void addPreviousOrder(Product orderProduct, int customerID) {
    int flag = 0;

    for(int i=0; i<customers.size(); i++) {
        if(customers.get(i).getID() == customerID) {
            customers.get(i).previousOrder.add(orderProduct);
            System.out.println("The product added to the past sales list");
            flag++;
        }
    }

    if(flag == 0)
        System.out.println("Product is already added in previous order list!");

}
```

T(n) = O(n^2 x m)

n: number of customer

m: number of previous order

## 14. Access the past order:

```java
public void accessThePastOrder(Customer customerObj) {
    for(int i=0; i<customers.size(); i++) {
        if(customers.get(i).getID() == customerObj.getID())
            customers.get(i).getPreviousOrder();
    }
}
```

T(n) = O(n^2 x m^2)

n: number of customer

m: number of previous order

15. Check stock:

```
public int getAllStockInfo(String productType) {
    int counter = 0;
    for(int i=0; i<allProducts.size(); i++) {
        if(allProducts.get(i).getType() == productType)
            counter++;
    }
    return counter;
}
```

T(n) = thea(n^2)