CSE 222/505 - Spring 2021 Homework 3

PART 2

Add branch:

```
public void add_branch(Branches Branch) {//

           Branchler.addLast(Branch);//O(1)

     }
```

$T(n)= \theta(1)$

Remove Branch:

```
public void remove(Branches branch) {
           Branchler.remove(branch);//  ➔ O(n)
     }
```

```
public boolean remove(E data) {
        int removeIndex = indexOf(data); ➔ O(1)
        if (removeIndex != -1) { ➔ O(1)
            remove(removeIndex);  ➔ O(n)
            return true;   ➔ O(1)
        }
        return false;    ➔ O(1)
    }

public E remove(int index) {
    if (index < 0 || index >= size) { ➔ O(1)
      throw new ArrayIndexOutOfBoundsException(index);
    }
    E returnValue = theData[index]; ➔ O(1)
    for (int i = index + 1; i < size; i++) { ➔ O(n)
      theData[i - 1] = theData[i];
    }
    size--;  ➔ O(1)
    return returnValue; ➔ O(1)
  }
```

$T(n)= \theta(n)$

Add Employee:

```
public void add_emp(BranchEmployee employee1) {

     employee.add(employee1); ➔ O(1)
}
```

```
public boolean add(E anEntry) {
    if (size == capacity) {
      reallocate();
    }
    theData[size] = anEntry; ➔ Amortized Constant Time
    size++;
    return true;
}
```

$T(n)= \theta(1)$

toString Employee:

```
public void display_emp() {

          for (int i = 0; i < employee.size(); i++) {
               System.out.println(i + "->" + 
employee.get(i).getName()+" " + employee.get(i).getSurname()+" "+ 
employee.get(i).getBranch_name() );
}
```

$T(n)= \theta(1)$

Remove Employee:

```
    public void remove_emp(String Branch_name, String rname, String
rsurname) {
        int flag = 0; ➔ O(1)
        for (int i = 0; i < employee.size(); i++) { ➔ O(n)

            if (Branch_name == employee.get(i).getBranch_name()
&& employee.get(i).getName() == rname&& employee.get(i).getSurname()
== rsurname) { ➔ O(1)
                flag = 1;
                getEmployee().remove(i); ➔ O(n)
            }

        }
        if (flag == 0) {
            System.out.println("Employee not found!"); ➔ O(1)
        } else {
            System.out.println("Employee is removed by admin");
➔ O(1)
        }

    }
```

T(n)= θ(n^2)

Customer:

Subscribes:

```
	public void Subscribes(Customer Custom1) {

		customers.add(Custom1);   ➔ O(1)
	}
```

Branch Employee:

Add Product:

```
	public void add_product(int index,String Type, String Model ,
String Color) {
		Furnitures.add(index, new Products(Type, Model, Color));
➔ O(n)
	}
```

```
public void add(int index, E item) {  ➔ O(1)
		int i = 0;
		if (index < 0 || index > list_size) {➔ O(1)
			throw new
IndexOutOfBoundsException(Integer.toString(index)); ➔ O(1)
		} else if (index == 0) { ➔ O(1)
			addFirst(item); ➔ O(1)
		} else {
			Node<E> node = head; ➔ O(1)
			while (i != (index - 1)) { ➔ O(n)
				node = node.next;
				i++;
			}
			addLast(node, item); ➔ O(1)
		}
	}
```

$T(n) = \theta(n)$