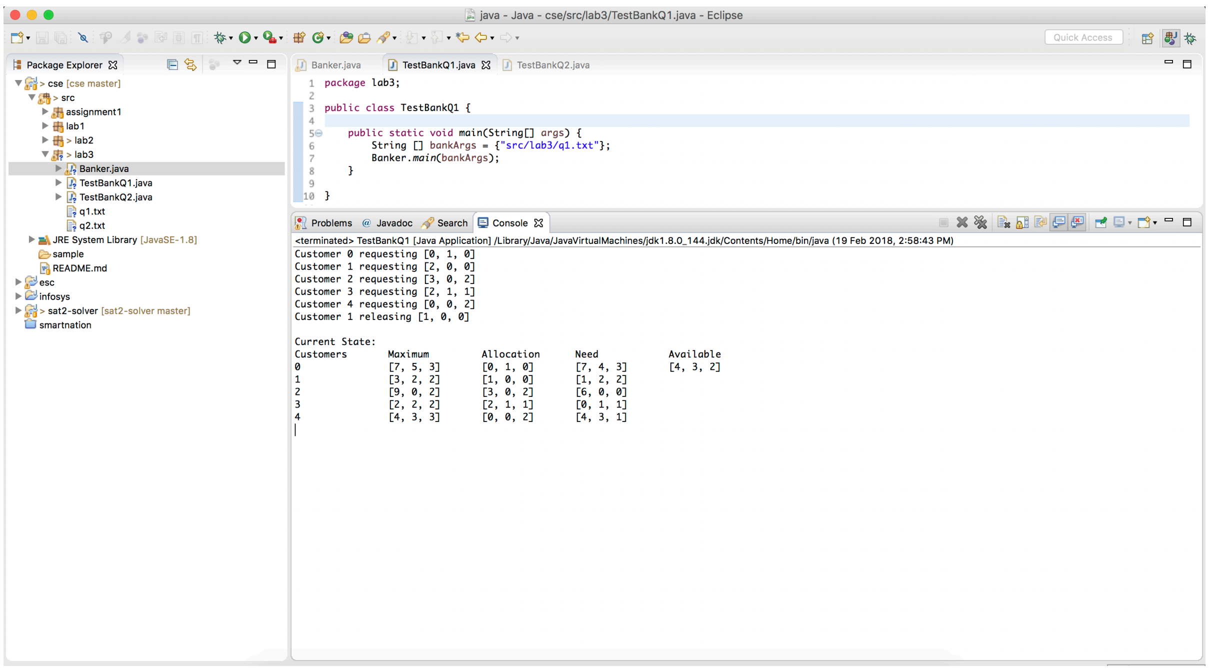
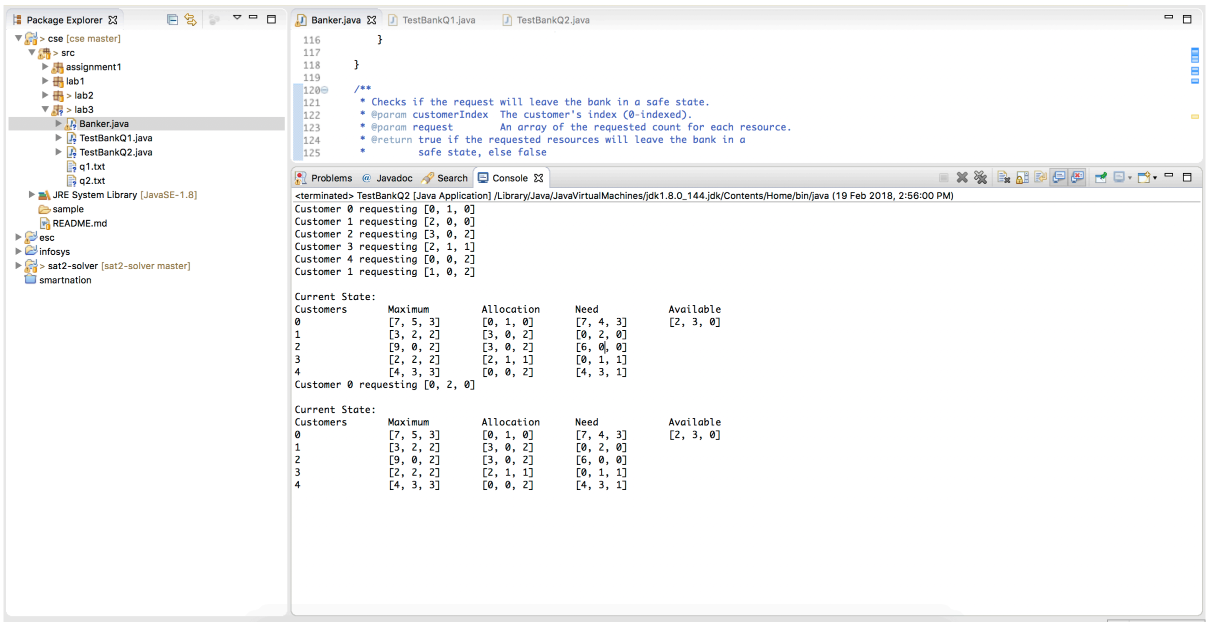
Lim Yao Jie
1001997
Lab3



```java
package lab3;

public class TestBankQ1 {

    public static void main(String[] args) {
        String [] bankArgs = {"src/lab3/q1.txt"};
        Banker.main(bankArgs);
    }

}
```

```
<terminated> TestBankQ1 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (19 Feb 2018, 2:58:43 PM)
Customer 0 requesting [0, 1, 0]
Customer 1 requesting [2, 0, 0]
Customer 2 requesting [3, 0, 2]
Customer 3 requesting [2, 1, 1]
Customer 4 requesting [0, 0, 2]
Customer 1 releasing [1, 0, 0]

Current State:
Customers       Maximum         Allocation      Need            Available
0               [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [4, 3, 2]
1               [3, 2, 2]       [1, 0, 0]       [1, 2, 2]
2               [9, 0, 2]       [3, 0, 2]       [6, 0, 0]
3               [2, 2, 2]       [2, 1, 1]       [0, 1, 1]
4               [4, 3, 3]       [0, 0, 2]       [4, 3, 1]
```



```java
        }

    }

    /**
     * Checks if the request will leave the bank in a safe state.
     * @param customerIndex  The customer's index (0-indexed).
     * @param request        An array of the requested count for each resource.
     * @return true if the requested resources will leave the bank in a
     *          safe state, else false
```

```
<terminated> TestBankQ2 [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (19 Feb 2018, 2:56:00 PM)
Customer 0 requesting [0, 1, 0]
Customer 1 requesting [2, 0, 0]
Customer 2 requesting [3, 0, 2]
Customer 3 requesting [2, 1, 1]
Customer 4 requesting [0, 0, 2]
Customer 1 requesting [1, 0, 2]

Current State:
Customers       Maximum         Allocation      Need            Available
0               [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [2, 3, 0]
1               [3, 2, 2]       [3, 0, 2]       [0, 2, 0]
2               [9, 0, 2]       [3, 0, 2]       [6, 0, 0]
3               [2, 2, 2]       [2, 1, 1]       [0, 1, 1]
4               [4, 3, 3]       [0, 0, 2]       [4, 3, 1]
Customer 0 requesting [0, 2, 0]

Current State:
Customers       Maximum         Allocation      Need            Available
0               [7, 5, 3]       [0, 1, 0]       [7, 4, 3]       [2, 3, 0]
1               [3, 2, 2]       [3, 0, 2]       [0, 2, 0]
2               [9, 0, 2]       [3, 0, 2]       [6, 0, 0]
3               [2, 2, 2]       [2, 1, 1]       [0, 1, 1]
4               [4, 3, 3]       [0, 0, 2]       [4, 3, 1]
```

Banker algorithm complexity

The majority of the complexity lies in the safety check algorithm. The complexity of the inner for loops are O(m*n), given n customers and m resources. The outer while loop runs at a worst case time of O(n) meaning the code runs ultimately at O(n*n*m).