

Analyze time complexity of Banker's Algorithm if

1. There is n customer in total
2. There is m resources in total

The Banker's algorithm time complexity will be dependent on the largest time complexity which is found in the check safe algorithm. It is when the process requests an available resource where the system must decide if granting the request will leave the system in a safe state.

```
while (possible) {
    possible = false;
    → for (int i = 0; i < numberOfCustomers; i++) {
        toWork = true;
        → for (int z = 0; z < numberOfResources; z++) {
            if (temp_need[i][z] > work[z])
                toWork = false;
        }
        if (finish[i] == false && toWork) {
            possible = true;
            → for (int j = 0; j < numberOfResources; j++) {
                work[j] += temp_allocation[i][j];
            }
            finish[i] = true;
        }
    }
}
```

The first for loop checks the number of customers $O(n)$, then it looks for the number of resources to check if the need is lesser more than the work value $O(m)$. If both conditions are met, i.e. the job is completed, and the need is less than the work it will update the work array, time complexity $O(m)$ times.

The rest of the codes are executed in for loops, but it has a maximum of only $O(n)$ or $O(m)$ respectively.

```
public void setMaximumDemand(int customerIndex, int[] maximumDemand) {
    // TODO: add customer, update maximum and need
    maximum[customerIndex] = maximumDemand; // customer indexes
    for (int i = 0; i < numberOfResources; i++) { // need n by m matrix, need[i,j] = k
        need[customerIndex][i] = maximumDemand[i] - allocation[customerIndex][i];
    }
}
```

Therefor the total time complexity for the Banker's Algorithm is **$O(n*m*m)$**

Screenshot Q2:

```
tengfone@LAPTOPGENESIS:~/50005Lab2/BankersAlgorithmLab/StarterCode_Java$ java TestBankQ2 q2_1.txt
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:
Available:
[2, 3, 0]

Maximum:
[7, 5, 3]
[3, 2, 2]
[9, 0, 2]
[2, 2, 2]
[4, 3, 3]

Allocation:
[0, 1, 0]
[3, 0, 2]
[3, 0, 2]
[2, 1, 1]
[0, 0, 2]

Need:
[7, 4, 3]
[0, 2, 0]
[6, 0, 0]
[0, 1, 1]
[4, 3, 1]

Customer 0 requesting
[0, 2, 0]

Current state:
Available:
[2, 3, 0]

Maximum:
[7, 5, 3]
[3, 2, 2]
[9, 0, 2]
[2, 2, 2]
[4, 3, 3]

Allocation:
[0, 1, 0]
[3, 0, 2]
[3, 0, 2]
[2, 1, 1]
[0, 0, 2]

Need:
[7, 4, 3]
[0, 2, 0]
[6, 0, 0]
[0, 1, 1]
[4, 3, 1]

tengfone@LAPTOPGENESIS:~/50005Lab2/BankersAlgorithmLab/StarterCode_Java$
```