

Q1

```

make && ./q1 testcases/q1_1.txt
gcc -o checkq1 C_code/checkerQ1.c
gcc -o q1 C_code/BankerQ1.c
gcc -o checkq2 C_code/checkerQ2.c
gcc -o q2 C_code/BankerQ2.c
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:
Available:
4 3 2
Maximum:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3

Allocation:
0 1 0
1 0 0
3 0 2
2 1 1
0 0 2

Need:
7 4 3
2 2 2
6 0 0
0 1 1
4 3 1

```

Q2

```

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

```

Q3

The algorithm goes through 1 loop for every customer (n loops).

Within each loop, it goes through n indexes for finish[], and within that, m loops to compare the values.

Overall complexity is hence  $O(n*n*m)$ .