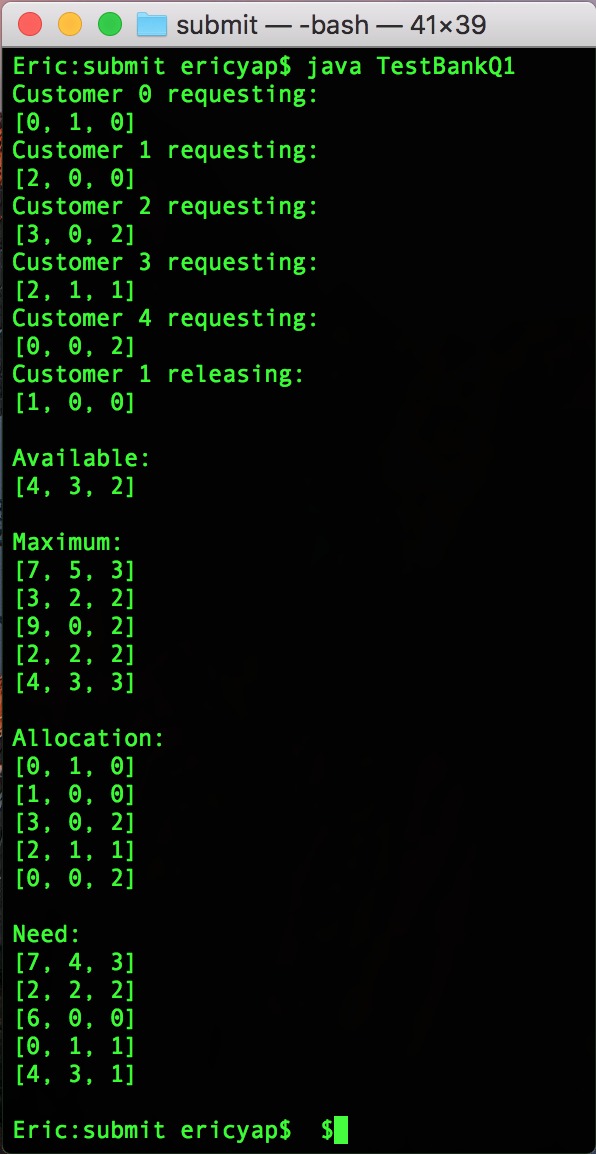
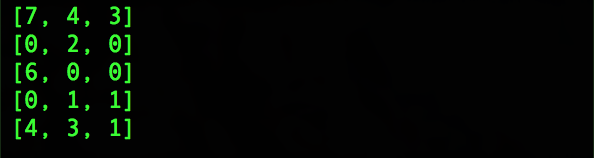
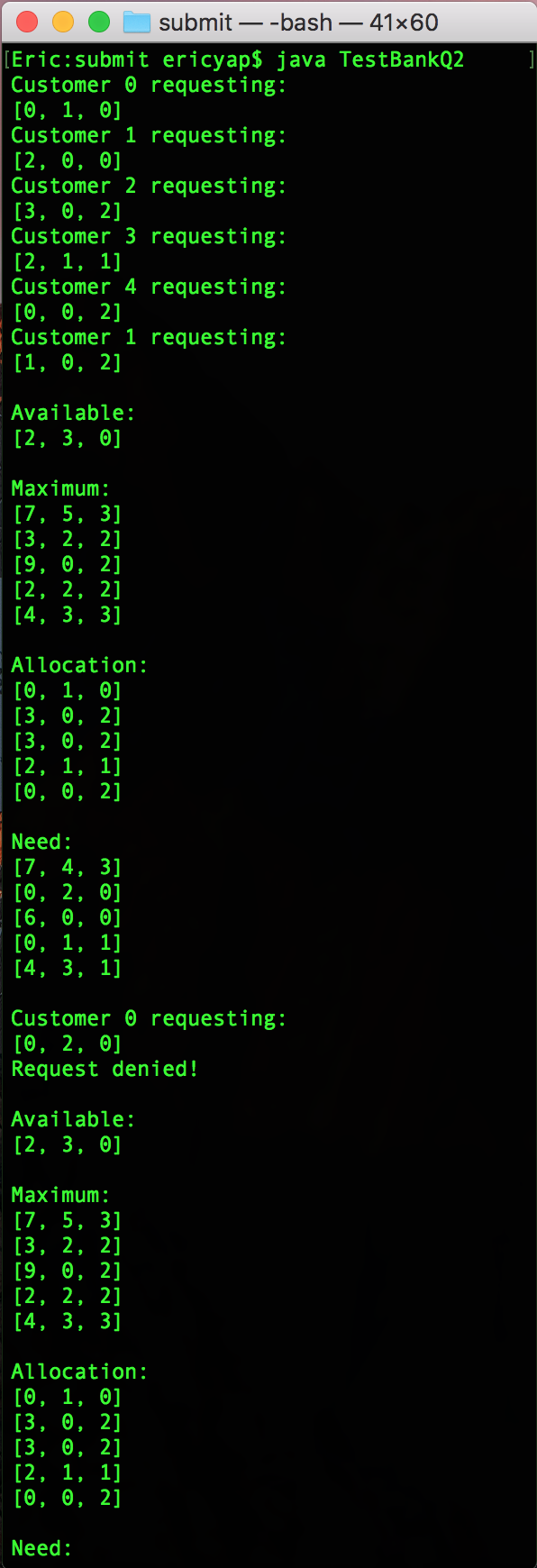
**50.005 Computer System Engineering Lab 3 Banker’s algorithm**

**Yap Wei Lok 1002394**

Screenshots



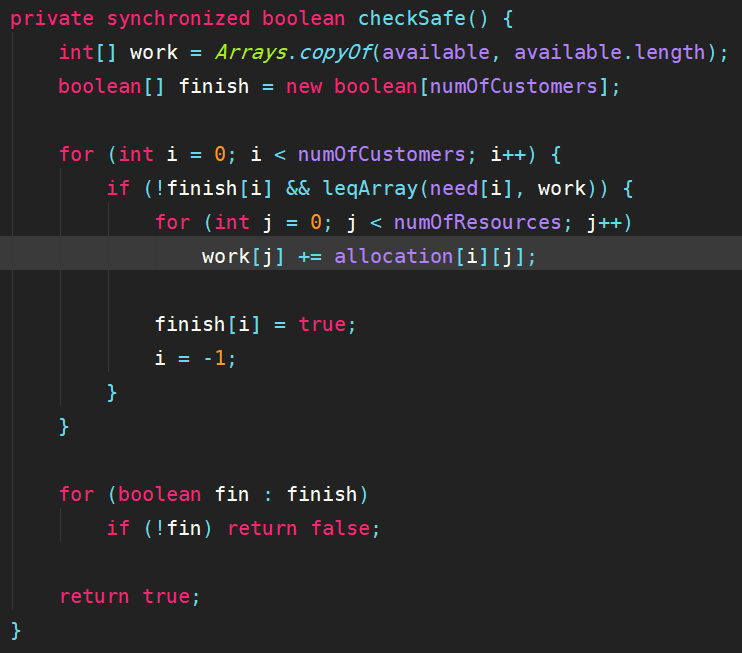


The screenshot on the left shows the output for q1 while the one on the right displays the output for q2.

On the second result, customer 0 requested [0, 2, 0] but was denied because no safe state can be found. Therefore, the available resources, maximum demanded resources, allocation and needed resources remain the same after the request.

Analysis

Let be the number of customers and be the number of resources.



Looking at the checkSafe function, there is a nested for loop in the function, the outer loop being since it goes through the and the inner loop being , going from to . The leqArray function also has complexity since it loops through the two arrays which are long to compare the array elements. The complexity so far is or to be short.

However, every time the execution goes into the <if> statement, the counting integer gets reset to -1. This means that in a worst-case scenario, the outer loop will go through . Therefore, the whole control flow has complexity.

There is also another loop that checks for the finish status of all customers, so the complexity is to be more specific, but suffice to describe the function’s complexity. The checkSafe function is also embedded in the requestResources function but is not nested inside any loop so it does not change the overall complexity of the Banker’s algorithm.

**Overall complexity:**