Sim Jia Ren (1004401) 50.005 Lab 2 Report

**Q3 On complexity**

The time complexity of the Banker’s algorithm is O(mn2), where n and m are the numbers of customers and resources respectively.

1. The allocation of temporary memory to copy the bank state takes a complexity of O(1) in checkSafe() function.
2. The copying of the bank’s state to the temporary memory takes a complexity of O(mn) due to the execution of two nested for-loops, iterating through the number of customers and resources.
3. The process of updating the temporary memory with the request takes a complexity of O(m) due to the execution of for-loop, iterating through the number of resources.
4. The process of setting the whole isFinished array to false takes a complexity of O(n).
5. In while (isPossible), the first iteration of the number of customers takes O(n) and followed by two iterations of the number of resources takes O(2m) = O(m). Overall, it takes O(mn).
   1. However, for the worst case scenario, it will take O(mn2). If there is one customer[i]’s isFinish[i] is set to be true, it has to run the while-loop n times for each customer.
6. Lastly, it checks for the whole isFinished array to be true, which takes a complexity of O(n).

Total time complexity = O(1 + mn + m + n + mn2 + n) = O(mn2)

**Result Screenshots**

Q1



Q2

