Lab 3 — Discussion

Baker's algorithm is centred around the checkSafe() method in our BankImpl class.

This safety algorithm first initialises it's vectors, available, maximum, allocation and need. The space complexity from this is O(n*m), where *n* is the number of processes and *m* is the number of resources.

This initialisation in the checkSafe() method also uses for-loops to set the values for each of the vectors, temp_avail, temp_need, temp_allocation, work and finish, taking a time complexity of O(n*m).

The next step in this algorithm checks for an unfinished process that could be completed with the currently available resources (a.k.a Finish[i] = false && Need[i] <= Work). This checking process can finish with Ω(n) if all processes do not satisfy the Need[i] <= Work condition. However, if every process is unfinished and all the needed resources can be satisfied by the available resources, the time complexity for this step will be O(m*m*n).

In the final step of checkSafe(), it is checked if every process is completed, taking a time complexity of O(n) and Ω(1).

However, there is a case where the very first resource requested by the customer is greater than the Need, allowing the entire checkSafe() to be bypassed. This allows the best case time complexity to be reduced to Ω(1).

In conclusion, the space complexity of Banker's algorithm is O(n*m), and the time complexity is O(m*m*n) and Ω(1).