1. Bankers’ algorithm
2. Deadlock detection algorithm

**Banker’s Algorithm**

Banker’s Algorithm is a**deadlock avoidance algorithm**. It is also used for deadlock detection. This algorithm tells that if any system can go into a deadlock or not by analyzing the currently allocated resources and the resources required by it in the future. There are various data structures which are used to implement this algorithm. So, let's learn about these first.

1. Available, it is a 1-D array that tells the number of each resource type currently available.
2. Max, an array that tells the maximum number of each resource type required by a process for successful execution.
3. Allocation, an array that tells the number of types of each resource type that has been allocated to the process.
4. Need, an array that tells the number of remaining instances of each resource type required for execution.

It consists of three steps to execute the banker’s algorithm in determining the safe state.

1. Generating the Need Matrix (Max – Allocation = Need)
2. Determining the safe sequence (Available + Allocation = New available)

Compare if the available array is greater than or equal to the current need. If it is, then add the allocation of the current need’s to available then continue the cycle till the end.

Example 1:

A picture containing text

Description automatically generated

A picture containing text

Description automatically generated

Example 2:

Table

Description automatically generated

Solution:

Text

Description automatically generated

Text

Description automatically generated

Example 1:

Text, table

Description automatically generated



Solution:

R = [3 4 2]

P3, P4

Example 2:

Table

Description automatically generated

Find R as well as determine the processes that are deadlocked.

Solution:

R = [7 2 6]

There is no deadlock in this example.