Detection Algorithm

1. Let ***Work*** and ***Finish*** be vectors of length ***m*** and ***n***, respectively
   Initialize:
   (a) ***Work*** = ***Available***
   (b) For i = **1,2, …, *n***, if $Allocation_i \neq 0$, then
   Finish*[i]** = ***false; otherwise, ***Finish*[i]** = ***true***

2. Find an index i such that both:
   (a) ***Finish*[*i*]** == ***false***
   (b) $Request_i \leq Work$

   If no such i exists, go to step 4

This means that either all the processes are finished, or all the unfinished processes are in need of more resource instances, thus made to wait for them to be available.

3. ***Work*** = ***Work*** + $Allocation_i$
   Finish*[*i*]** = ***true
   go to step 2

Algorithm requires an order of $O(m \times n^2)$ operations to detect whether the system is in deadlocked state

4. If ***Finish[i]*** == ***false***, for some i, $1 \leq i \leq n$, then the system is in deadlock state. Moreover, if ***Finish*[*i*]** == ***false*** for an i, then P_i is deadlocked