CECS 526 Assignment 7 (2 points) **Name: Varun Lingabathini__**
Due : April 21, 2020, by class time on BeachBoard

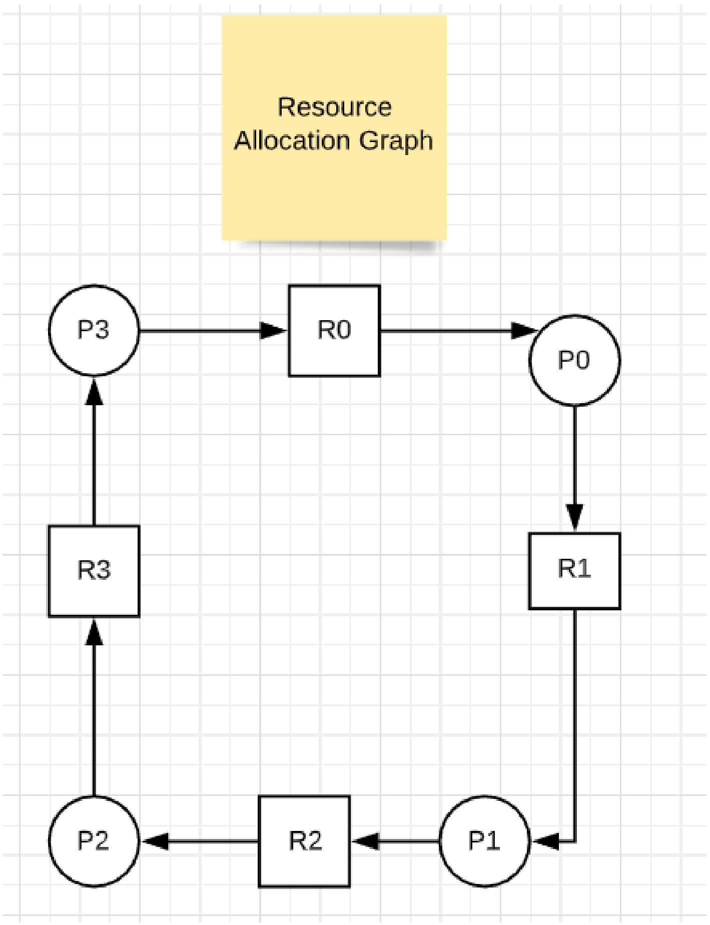
Given the following scheme for distributed deadlock detection:

Processes are assigned unique priorities. When a process waits for an object being locked by another process, deadlock detection is initiated and a probe is sent out to the process where the object is locked. When a probe arrives at an unblocked process or a blocked process with priority higher than the probe initiator, the probe is discarded; otherwise the probe is forwarded along the wait-for edge. A deadlock is declared when a probe returns to the site where it is originated.

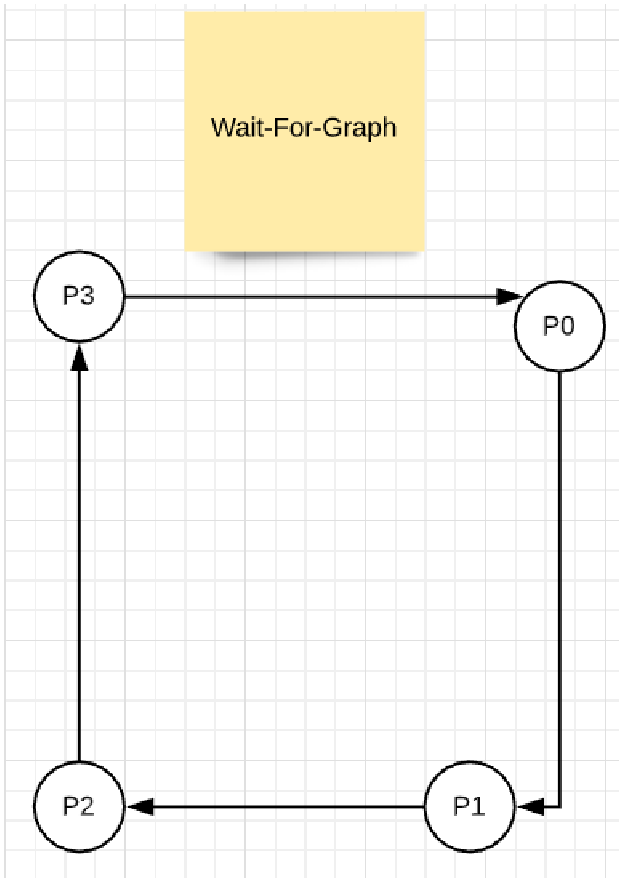
a) Given a scenario with 4 processes P_0, P_1, P_2, and P_3, and 4 resource types R_0, R_1, R_2, and R_3, where each process resides in a separate site and each resource type has a single instance. Currently, each R_i is assigned to P_i, for $0 \le i \le 3$. Processes are prioritized, with P_0 being the highest and P_3 the lowest.

Suppose at this time, each P_i is requesting for resource $R_{(i+1) \bmod 4}$, for $0 \le i \le 3$. (A deadlock cycle should have formed as a result.) These requests will trigger deadlock detection according to the above scheme. Follow the analysis steps below to determine the total number of messages transmitted in detecting this deadlock.

1. Construct the resource allocation graph of this system.



2. Convert the resource allocation graph into a corresponding wait-for-graph. Observe the formation of a cycle, which represents the existence of a deadlock.



3. Find the number of messages transmitted from each probe initiated, and compute the total.

The number of messages transmitted when P0 initiates deadlock detection is 4 likewise , P1 is 3, P2 is 2 and P3 is 1 . So the total number of messages transmitted would be 10.

b) Generalize the above scenario to *n* processes in *n* sites, with each process holding a separate resource and requesting for an assigned resource in the same pattern as described above, and determine the total number of messages transmitted for the detection of the existing deadlock.

In the scenario where n processes present in n sites , the highest priority process sends out n messages , the second highest process sends out n-1 messages and this happens until the last process.
Therefore the total messages required will be sum of *n, n-1..1* which equals *n(n+1)/2* .

c) Part b) above represents the worst case performance for the given deadlock detection scheme in terms of the total number of messages transmitted for detecting a deadlock formed under the following conditions: every site is running one process, resource requests by individual processes are limited to one resource at a time, and there is currently a deadlock cycle that involves all *n* processes. Find the performance of the scheme for a best case scenario with *n* processes.

The best case is when the highest priority process starts deadlock detection initiation and the deadlock is detected . So the total messages sent in this scenario would be *n*.