**CS425 2015 Spring: Distributed System**

# MP3: A Mutual Exclusion Algorithm

**Haitong Tian, NetID: htian3**

1: The nodes are indexed from 0 to 8. I aligned the nodes in a 3X3 matrix, where for node i, its voting set includes all nodes that are in the same row or column with itself.

For node 0, its voting set are [0, 1, 2, 3, 6]
For node 1, its voting set are [0, 1, 2, 4, 7]
For node 2, its voting set are [0, 1, 2, 5, 8]
For node 3, its voting set are [0, 3, 4, 5, 6]
For node 4, its voting set are [1, 3, 4, 5, 7]
For node 5, its voting set are [2, 3, 4, 5, 8]
For node 6, its voting set are [0, 3, 6, 7, 8]
For node 7, its voting set are [1, 4, 6, 7, 8]
For node 8, its voting set are [2, 5, 6, 7, 8]

2: Avoiding deadlocks: The nodes are ordered with different priorities, with 0 the lowest priority and 8 the highest priority. When requesting mutex, a node must get consent from node with smaller index before asking the node with larger index. By using ordered resources, deadlock can be avoided.

3: To compile the code, run “sh compile.sh” under the “CS425MP3” folder.

To run the code, go to “bin” directory, and run “java Mutex <cs_int> <next_req> <tot_exec_time> <option>”

For example, you can run “java Mutex 5 7 1 1”