# Bonus Report for Chords Algorithm

**Team Members:**

- Anurag Bagalwadi (UFID: 4936 9125)
- Fatema Saifee (UFID: 1508 1278)

## Instructions

**Expected Input**

- numNodes - the number of peers to be created in the peer to peer system
- numRequests - the number of requests each peer has to make.

Report For Bonus is present in Project3-Bonus Folder

## Sample

**Input**

mix run lib/chords.exs <numberOfNodes> <numberOfRequests>

For input numberOfNodes = 50, numberOfRequests = 1

*'mix run lib/chords.exs 50 1'*

**Output**

*T_____*
*Node:*
*{#PID<0.137.0>,*
*"12DE0ECAF7A6AB5E75FCCA6A2E711F7BB29B2C751661D2441E719FBD35BAE144"}*
*died*

*_____*

*Stabilizing network...*

*Reassigning keys of dead node...*

*Updating all finger tables...*

*The average number of hops (node connections) that have to be traversed to deliver a message is 7*

# Implementation

An additional state is added to each Node Process call its Status which can have 2 possible value

- **:active** - Node has failed
- **:dead** - Node is alive

The methods used are as follows:

1. **killRandomNode(**pidHashMap**)**
   Choses a node from <pidHashMap> list and updates its status in the state from :active to :dead. It returns the dead Node. Arguments are as follows:
   - pidHashMap - {PID, hashedPID} list Sorted on hashedPIDs

   Example

   *iex> Chords.killRandomNode(*pidHashMap*)*

   Output

   *{#PID<0.122.0>, "1A2EF8ADECC2BB0CF46A7E192A015C371C9D2B4902986205D0DABDCA98D431D7"}*
2. **Stabilize(**pidHashMap, deadNode**)**
   It call two other funtions to update the keys and fingers of other active nodes of the network. Arguments are as follows:
   - pidHashMap - [{PID, hashedPID}] list Sorted on hashedPIDs
   - deadNode - {PID, hashedPID} of the node whose status is :dead
3. **updateKeysOfDeadNode(**pidHashMap)
   This assigns the keys of the dead node to its successor
4. **updateFingerTables(**pidHashMap, deadNod**)**
   Updates the fingers of all the nodes in the chord going anti-clockwise. Arguments are as follows:
   - pidHashMap - [{PID, hashedPID}] list Sorted on hashedPIDs
   - deadNode - {PID, hashedPID} of the node whose status is :dead