chord-protocol

Implemented the famous chord protocol described in https://pdos.csail.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf

Team member: Harish Reddy Bollavaram

How to run the program

```
dotnet fsi Project3.fsx <number-of-nodes> <number-of-requests>
```
Example,
```
dotnet fsi Project3.fsx 1024 16
```

What is working

1. All APIs specified in the paper like `join` , `find_successor`, `stabilize` and optimizations like `finger_table` are working.
2. `join` and `stabilize` API are merged into a single `join` to reduce complexity. The `join` operation had to be run sequentially in order to avoid having to run stabilization periodically.
3. This implementation cannot handle node failures, but can handle node addition.
4. Each request takes atmost logN hops where N is the number of nodes.
5. Each node has a finger table of size logN

Largest network

Nodes	**1024**	**4000**	**8000**
Hops(avg)	4.2	4.8	5.2

My lookup is super fast but the join takes too long for large networks because it has to be sequential