

# Project 3: P2P

## Team members

- Vishisht Khilariwal
- Siddhant Jain

Command to Run : **dotnet fsi Chord.fsx 100 100**

## What is working

- Actor Nodes are created and then joined to each other using the Join API defined in the research paper
- Every actor stabilises itself periodically (every 1 sec) to fix its neighbours i.e the predecessors and the successors with the logic defined in the paper
- Every actor fixes it's finger table to enable efficient routing in  $\log(N)$  steps in worst case by running fix finger API defined in the paper every second
- Nodes send requests to find a key in the system

When the system is not stable the finger tables are still being fixed. The messages sent at that time take longer to route and sometimes may not be delivered since all the nodes might not have been discovered.

After the System stabilises and the finger tables are also stable the routing becomes efficient

Below is the screenshot for 80 nodes equally distributed in the network represented in 8 bits( Max keys = 256)

```
TotalHops: 851788, Total Requests: 742, Average Hops: 1147.962264, Efficient Hops: 2497, Efficient Requests 691, Efficient Average Hops: 3.613603473
TotalHops: 851791, Total Requests: 743, Average Hops: 1146.421265, Efficient Hops: 2500, Efficient Requests 692, Efficient Average Hops: 3.612716763
TotalHops: 851794, Total Requests: 744, Average Hops: 1144.884409, Efficient Hops: 2503, Efficient Requests 693, Efficient Average Hops: 3.611832612
TotalHops: 851797, Total Requests: 745, Average Hops: 1143.351678, Efficient Hops: 2506, Efficient Requests 694, Efficient Average Hops: 3.610951009
TotalHops: 851800, Total Requests: 746, Average Hops: 1141.823056, Efficient Hops: 2509, Efficient Requests 695, Efficient Average Hops: 3.610071942
TotalHops: 851803, Total Requests: 747, Average Hops: 1140.298527, Efficient Hops: 2512, Efficient Requests 696, Efficient Average Hops: 3.609195402
TotalHops: 851806, Total Requests: 748, Average Hops: 1138.778075, Efficient Hops: 2515, Efficient Requests 697, Efficient Average Hops: 3.608321377
TotalHops: 851809, Total Requests: 749, Average Hops: 1137.261682, Efficient Hops: 2518, Efficient Requests 698, Efficient Average Hops: 3.607449857
TotalHops: 851812, Total Requests: 750, Average Hops: 1135.749333, Efficient Hops: 2521, Efficient Requests 699, Efficient Average Hops: 3.606580883
TotalHops: 851816, Total Requests: 751, Average Hops: 1134.242344, Efficient Hops: 2525, Efficient Requests 700, Efficient Average Hops: 3.607142857
TotalHops: 851820, Total Requests: 752, Average Hops: 1132.739362, Efficient Hops: 2529, Efficient Requests 701, Efficient Average Hops: 3.607703281
TotalHops: 851824, Total Requests: 753, Average Hops: 1131.240372, Efficient Hops: 2533, Efficient Requests 702, Efficient Average Hops: 3.608262108
TotalHops: 851828, Total Requests: 754, Average Hops: 1129.745358, Efficient Hops: 2537, Efficient Requests 703, Efficient Average Hops: 3.608819346
TotalHops: 851830, Total Requests: 755, Average Hops: 1128.251656, Efficient Hops: 2539, Efficient Requests 704, Efficient Average Hops: 3.606534091
TotalHops: 851833, Total Requests: 756, Average Hops: 1126.763228, Efficient Hops: 2542, Efficient Requests 705, Efficient Average Hops: 3.605673759
TotalHops: 851836, Total Requests: 757, Average Hops: 1125.278732, Efficient Hops: 2545, Efficient Requests 706, Efficient Average Hops: 3.604815864
TotalHops: 851839, Total Requests: 758, Average Hops: 1123.798153, Efficient Hops: 2548, Efficient Requests 707, Efficient Average Hops: 3.603960396
TotalHops: 851842, Total Requests: 759, Average Hops: 1122.321476, Efficient Hops: 2551, Efficient Requests 708, Efficient Average Hops: 3.603107345
TotalHops: 851845, Total Requests: 760, Average Hops: 1120.848684, Efficient Hops: 2554, Efficient Requests 709, Efficient Average Hops: 3.6022567
TotalHops: 851848, Total Requests: 761, Average Hops: 1119.379763, Efficient Hops: 2557, Efficient Requests 710, Efficient Average Hops: 3.601408451
TotalHops: 851852, Total Requests: 762, Average Hops: 1117.91601, Efficient Hops: 2561, Efficient Requests 711, Efficient Average Hops: 3.601969058
TotalHops: 851856, Total Requests: 763, Average Hops: 1116.456094, Efficient Hops: 2565, Efficient Requests 712, Efficient Average Hops: 3.60252809
TotalHops: 851860, Total Requests: 764, Average Hops: 1115.0, Efficient Hops: 2569, Efficient Requests 713, Efficient Average Hops: 3.603085554
```

Here efficient requests are defined as the requests which took less than 8 (no. of bits) hops to reach the target.

As we can see for this network 713 requests were efficient out of 764 total requests. The average hops to reach the target for the efficient requests were 3.6

## What is the largest network you managed to deal with

For 500 Nodes and 15 bits network avg hops are 5.4

```
33
34 let r = Random() // Random
35
36 let generateRequest() = // unit -> int
PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE
TotalHops: 74087, Total Requests: 2701, Average Hops: 27.42947057, Efficient Hops: 10882, Efficient Requests 1336, Efficient Average Hops: 7.546407186
TotalHops: 74467, Total Requests: 2801, Average Hops: 26.58586219, Efficient Hops: 10462, Efficient Requests 1436, Efficient Average Hops: 7.28551532
TotalHops: 75124, Total Requests: 2901, Average Hops: 25.89589797, Efficient Hops: 11119, Efficient Requests 1536, Efficient Average Hops: 7.238932292
TotalHops: 75813, Total Requests: 3001, Average Hops: 25.26257914, Efficient Hops: 11088, Efficient Requests 1636, Efficient Average Hops: 7.217603912
TotalHops: 76124, Total Requests: 3101, Average Hops: 24.54281025, Efficient Hops: 12119, Efficient Requests 1736, Efficient Average Hops: 6.980990783
TotalHops: 76429, Total Requests: 3201, Average Hops: 23.87660106, Efficient Hops: 12424, Efficient Requests 1836, Efficient Average Hops: 6.766884532
TotalHops: 76852, Total Requests: 3301, Average Hops: 23.28142987, Efficient Hops: 12847, Efficient Requests 1936, Efficient Average Hops: 6.635847107
TotalHops: 77294, Total Requests: 3401, Average Hops: 22.72684505, Efficient Hops: 13289, Efficient Requests 2036, Efficient Average Hops: 6.527013752
TotalHops: 77862, Total Requests: 3501, Average Hops: 22.23993145, Efficient Hops: 13857, Efficient Requests 2136, Efficient Average Hops: 6.487359551
TotalHops: 78191, Total Requests: 3601, Average Hops: 21.71369664, Efficient Hops: 14186, Efficient Requests 2236, Efficient Average Hops: 6.344364937
TotalHops: 78516, Total Requests: 3701, Average Hops: 21.21480681, Efficient Hops: 14511, Efficient Requests 2336, Efficient Average Hops: 6.211900685
TotalHops: 78851, Total Requests: 3801, Average Hops: 20.744804, Efficient Hops: 14846, Efficient Requests 2436, Efficient Average Hops: 6.094417077
TotalHops: 79264, Total Requests: 3901, Average Hops: 20.31889259, Efficient Hops: 15259, Efficient Requests 2536, Efficient Average Hops: 6.016955836
TotalHops: 79744, Total Requests: 4001, Average Hops: 19.93101725, Efficient Hops: 15739, Efficient Requests 2636, Efficient Average Hops: 5.970789974
TotalHops: 80102, Total Requests: 4101, Average Hops: 19.53230919, Efficient Hops: 16097, Efficient Requests 2736, Efficient Average Hops: 5.883406433
TotalHops: 80429, Total Requests: 4201, Average Hops: 19.14520352, Efficient Hops: 16424, Efficient Requests 2836, Efficient Average Hops: 5.791255289
TotalHops: 80776, Total Requests: 4301, Average Hops: 18.78074866, Efficient Hops: 16771, Efficient Requests 2936, Efficient Average Hops: 5.71219346
TotalHops: 81158, Total Requests: 4401, Average Hops: 18.44980891, Efficient Hops: 17153, Efficient Requests 3036, Efficient Average Hops: 5.649868248
TotalHops: 81614, Total Requests: 4501, Average Hops: 18.13241502, Efficient Hops: 17609, Efficient Requests 3136, Efficient Average Hops: 5.615114796
TotalHops: 81895, Total Requests: 4601, Average Hops: 17.79939144, Efficient Hops: 17890, Efficient Requests 3236, Efficient Average Hops: 5.528430161
TotalHops: 82244, Total Requests: 4701, Average Hops: 17.49590106, Efficient Hops: 18239, Efficient Requests 3336, Efficient Average Hops: 5.467326139
TotalHops: 82656, Total Requests: 4801, Average Hops: 17.21641325, Efficient Hops: 18651, Efficient Requests 3436, Efficient Average Hops: 5.428114086
TotalHops: 286692, Total Requests: 4901, Average Hops: 58.4726974, Efficient Hops: 19194, Efficient Requests 3535, Efficient Average Hops: 5.42970297

```

For 1000 Nodes and 20 bit network

```
31
32 let Chord(mailbox: Actor<_>) = // Actor<SupervisorMsg> -> Cont<SupervisorMsg, 'a>
33
34 PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE
35
36 TotalHops: 221901, Total Requests: 4501, Average Hops: 49.30037769, Efficient Hops: 9822, Efficient Requests 787, Efficient Average Hops: 12.48030496
TotalHops: 227669, Total Requests: 4601, Average Hops: 49.4825038, Efficient Hops: 9822, Efficient Requests 787, Efficient Average Hops: 12.48030496
TotalHops: 234365, Total Requests: 4701, Average Hops: 49.85428632, Efficient Hops: 9822, Efficient Requests 787, Efficient Average Hops: 12.48030496
TotalHops: 242190, Total Requests: 4801, Average Hops: 50.44574047, Efficient Hops: 9822, Efficient Requests 787, Efficient Average Hops: 12.48030496
TotalHops: 251746, Total Requests: 4901, Average Hops: 51.36625179, Efficient Hops: 9822, Efficient Requests 787, Efficient Average Hops: 12.48030496
TotalHops: 264184, Total Requests: 5001, Average Hops: 52.82623475, Efficient Hops: 9822, Efficient Requests 788, Efficient Average Hops: 12.46446791
TotalHops: 265269, Total Requests: 5101, Average Hops: 52.0033268, Efficient Hops: 10749, Efficient Requests 881, Efficient Average Hops: 12.20099806
TotalHops: 267728, Total Requests: 5201, Average Hops: 52.17298623, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 275520, Total Requests: 5401, Average Hops: 51.01277541, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 280586, Total Requests: 5501, Average Hops: 51.00636248, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 286580, Total Requests: 5601, Average Hops: 51.16586324, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 293448, Total Requests: 5701, Average Hops: 51.4730749, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 301543, Total Requests: 5801, Average Hops: 51.98121014, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 311327, Total Requests: 5901, Average Hops: 52.75834604, Efficient Hops: 11292, Efficient Requests 910, Efficient Average Hops: 12.40879121
TotalHops: 323865, Total Requests: 6001, Average Hops: 53.96856525, Efficient Hops: 11293, Efficient Requests 911, Efficient Average Hops: 12.39626784
TotalHops: 324613, Total Requests: 6101, Average Hops: 53.20652352, Efficient Hops: 12041, Efficient Requests 1011, Efficient Average Hops: 11.99999011
TotalHops: 326039, Total Requests: 6201, Average Hops: 52.57845509, Efficient Hops: 13197, Efficient Requests 1099, Efficient Average Hops: 12.00818926
TotalHops: 327939, Total Requests: 6301, Average Hops: 52.04554833, Efficient Hops: 14551, Efficient Requests 1178, Efficient Average Hops: 12.35229202
TotalHops: 330301, Total Requests: 6401, Average Hops: 51.60146852, Efficient Hops: 14842, Efficient Requests 1193, Efficient Average Hops: 12.449905528
TotalHops: 333244, Total Requests: 6501, Average Hops: 51.26042147, Efficient Hops: 14842, Efficient Requests 1193, Efficient Average Hops: 12.449905528
TotalHops: 336958, Total Requests: 6601, Average Hops: 51.0465081, Efficient Hops: 14842, Efficient Requests 1193, Efficient Average Hops: 12.449905528
TotalHops: 341025, Total Requests: 6701, Average Hops: 50.89165796, Efficient Hops: 14842, Efficient Requests 1193, Efficient Average Hops: 12.449905528

```

Running for 10,000 Nodes gave stack overflow error