

Chord

This document uses a **sample test case** to show the grading steps, my codes are written in Python.

If you find any problems, please email Yi Liu (yliu634@ucsc.edu)

1. Add nodes to the network using the join function, $m = 8$

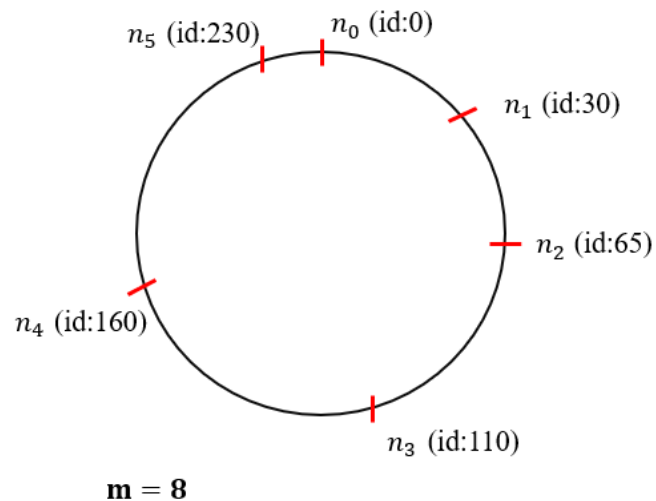


Fig.1 An identifier circle that contains six nodes

```
n0 = Node(0)
n1 = Node(30)
n2 = Node(65)
n3 = Node(110)
n4 = Node(160)
n5 = Node(230)
n0.join(None)
n1.join(n0)
n2.join(n1)
n3.join(n2)
n4.join(n3)
n5.join(n4)
```

2. Print finger table of all nodes (40pts)

<pre> -----Node id:0----- Successor: 30 Predecessor: 230 FingerTables: k = 1 [1 , 2) succ. = 30 k = 2 [2 , 4) succ. = 30 k = 3 [4 , 8) succ. = 30 k = 4 [8 , 16) succ. = 30 k = 5 [16 , 32) succ. = 30 k = 6 [32 , 64) succ. = 65 k = 7 [64 , 128) succ. = 65 k = 8 [128 , 0) succ. = 160 ----- ***** </pre>	<pre> -----Node id:30----- Successor: 65 Predecessor: 0 FingerTables: k = 1 [31 , 32) succ. = 65 k = 2 [32 , 34) succ. = 65 k = 3 [34 , 38) succ. = 65 k = 4 [38 , 46) succ. = 65 k = 5 [46 , 62) succ. = 65 k = 6 [62 , 94) succ. = 65 k = 7 [94 , 158) succ. = 110 k = 8 [158 , 30) succ. = 160 ----- ***** </pre>
<pre> -----Node id:65----- Successor: 110 Predecessor: 30 FingerTables: k = 1 [66 , 67) succ. = 110 k = 2 [67 , 69) succ. = 110 k = 3 [69 , 73) succ. = 110 k = 4 [73 , 81) succ. = 110 k = 5 [81 , 97) succ. = 110 k = 6 [97 , 129) succ. = 110 k = 7 [129 , 193) succ. = 160 k = 8 [193 , 65) succ. = 230 ----- ***** </pre>	<pre> -----Node id:110----- Successor: 160 Predecessor: 65 FingerTables: k = 1 [111 , 112) succ. = 160 k = 2 [112 , 114) succ. = 160 k = 3 [114 , 118) succ. = 160 k = 4 [118 , 126) succ. = 160 k = 5 [126 , 142) succ. = 160 k = 6 [142 , 174) succ. = 160 k = 7 [174 , 238) succ. = 230 k = 8 [238 , 110) succ. = 0 ----- ***** </pre>
<pre> -----Node id:160----- Successor: 230 Predecessor: 110 FingerTables: k = 1 [161 , 162) succ. = 230 k = 2 [162 , 164) succ. = 230 k = 3 [164 , 168) succ. = 230 k = 4 [168 , 176) succ. = 230 k = 5 [176 , 192) succ. = 230 k = 6 [192 , 224) succ. = 230 k = 7 [224 , 32) succ. = 230 k = 8 [32 , 160) succ. = 65 ----- ***** </pre>	<pre> -----Node id:230----- Successor: 0 Predecessor: 160 FingerTables: k = 1 [231 , 232) succ. = 0 k = 2 [232 , 234) succ. = 0 k = 3 [234 , 238) succ. = 0 k = 4 [238 , 246) succ. = 0 k = 5 [246 , 6) succ. = 0 k = 6 [6 , 38) succ. = 30 k = 7 [38 , 102) succ. = 65 k = 8 [102 , 230) succ. = 110 ----- ***** </pre>

Fig.2 Finger tables of six nodes

3. Insert keys and add new node joins (20pts)

```

n0.insert(3, 3)
n1.insert(200)
n2.insert(123)
n3.insert(45,3)
n4.insert(99)
n2.insert(60,10)
n0.insert(50,8)
n3.insert(100,5)
n3.insert(101,4)
n3.insert(102,6)
n5.insert(240,8)
n5.insert(250,10)

```

3.1 print keys that stored in each node (10pts)

```
-----Node id:0-----
{240: 8, 250: 10}
-----Node id:30-----
{3: 3}
-----Node id:65-----
{45: 3, 60: 10, 50: 8}
-----Node id:110-----
{99: None, 100: 5, 101: 4, 102: 6}
-----Node id:160-----
{123: None}
-----Node id:230-----
{200: None}
```

Fig.3 keys distribution on nodes

n_6 (id: 100) joins

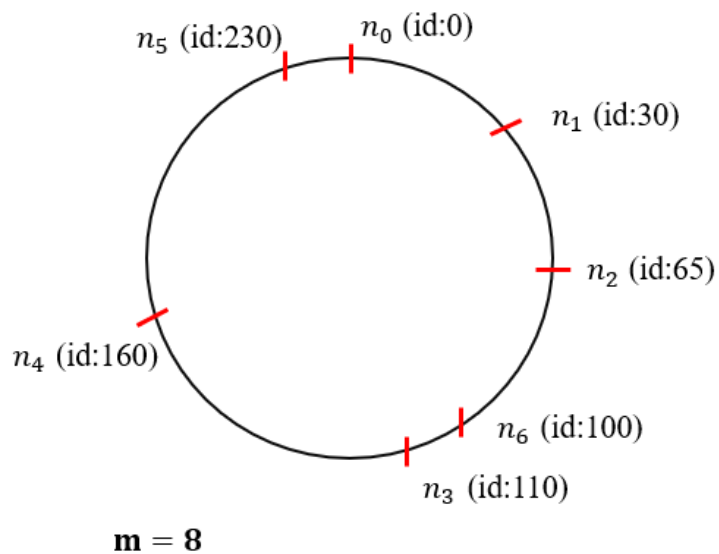


Fig.4 An updated circle after n_6 joins

3.2 Print migrated keys (10pts)

```
*****
migrate key 99 from node 110 to node 100
migrate key 100 from node 110 to node 100
```

4. Lookup keys (40pts)

Print lookup results and sequences of nodes get involved in this procedure (run lookup on node n_0 , n_2 , n_6 for all keys)

```

-----node 0-----
Look-up result of key 3 from node 0 with path [0,30] value is 3
Look-up result of key 200 from node 0 with path [0,230] value is None
Look-up result of key 123 from node 0 with path [0,160] value is None
Look-up result of key 45 from node 0 with path [0,65] value is 3
Look-up result of key 99 from node 0 with path [0,100] value is None
Look-up result of key 60 from node 0 with path [0,65] value is 10
Look-up result of key 50 from node 0 with path [0,65] value is 8
Look-up result of key 100 from node 0 with path [0,100] value is 5
Look-up result of key 101 from node 0 with path [0,110] value is 4
Look-up result of key 102 from node 0 with path [0,110] value is 6
Look-up result of key 240 from node 0 with path [0] value is 8
Look-up result of key 250 from node 0 with path [0] value is 10
-----

-----node 65-----
Look-up result of key 3 from node 65 with path [65,30] value is 3
Look-up result of key 200 from node 65 with path [65,230] value is None
Look-up result of key 123 from node 65 with path [65,160] value is None
Look-up result of key 45 from node 65 with path [65] value is 3
Look-up result of key 99 from node 65 with path [65,100] value is None
Look-up result of key 60 from node 65 with path [65] value is 10
Look-up result of key 50 from node 65 with path [65] value is 8
Look-up result of key 100 from node 65 with path [65,100] value is 5
Look-up result of key 101 from node 65 with path [65,110] value is 4
Look-up result of key 102 from node 65 with path [65,110] value is 6
Look-up result of key 240 from node 65 with path [65,0] value is 8
Look-up result of key 250 from node 65 with path [65,0] value is 10

-----node 100-----
Look-up result of key 3 from node 100 with path [100,30] value is 3
Look-up result of key 200 from node 100 with path [100,230] value is None
Look-up result of key 123 from node 100 with path [100,160] value is None
Look-up result of key 45 from node 100 with path [100,65] value is 3
Look-up result of key 99 from node 100 with path [100] value is None
Look-up result of key 60 from node 100 with path [100,65] value is 10
Look-up result of key 50 from node 100 with path [100,65] value is 8
Look-up result of key 100 from node 100 with path [100] value is 5
Look-up result of key 101 from node 100 with path [100,110] value is 4
Look-up result of key 102 from node 100 with path [100,110] value is 6
Look-up result of key 240 from node 100 with path [100,0] value is 8
Look-up result of key 250 from node 100 with path [100,0] value is 10
-----

```

Fig.5 Lookup keys from three nodes

5. Leave (20 pts)

Let one node **n2** leave, print the updated finger tables of n0 and n1, and keys

distribution (Please also show keys distribution on nodes).

```
-----Node id:0-----
Successor: 30 Predecessor: 230
FingerTables:
| k = 1 [ 1 , 2 )      succ. = 30 |
| k = 2 [ 2 , 4 )      succ. = 30 |
| k = 3 [ 4 , 8 )      succ. = 30 |
| k = 4 [ 8 , 16 )     succ. = 30 |
| k = 5 [ 16 , 32 )    succ. = 30 |
| k = 6 [ 32 , 64 )    succ. = 100 |
| k = 7 [ 64 , 128 )   succ. = 100 |
| k = 8 [ 128 , 0 )    succ. = 160 |
-----

-----Node id:30-----
Successor: 100 Predecessor: 0
FingerTables:
| k = 1 [ 31 , 32 )    succ. = 100 |
| k = 2 [ 32 , 34 )    succ. = 100 |
| k = 3 [ 34 , 38 )    succ. = 100 |
| k = 4 [ 38 , 46 )    succ. = 100 |
| k = 5 [ 46 , 62 )    succ. = 100 |
| k = 6 [ 62 , 94 )    succ. = 100 |
| k = 7 [ 94 , 158 )   succ. = 100 |
| k = 8 [ 158 , 30 )   succ. = 160 |
-----
*****
```

Fig.6 Updated finger table