a)

b)

Node 2

start	interval	succ
3	(3,4)	4
4	(4,6)	6
6	(6,10)	10
10	(10,3)	13

Node 4

start	interval	succ
5	(5,6)	6
6	(6,8)	10
8	(8,12)	10
12	(12,5)	13

Node 6

start	interval	succ
7	(7,8)	10
8	(8,10)	10
10	(10,14)	13
14	(14,6)	2

Node 10

start	interval	succ
11	(11,12)	13
12	(12,14)	13
14	(14,12)	2
2	(2,10)	4

Node 13

start	interval	succ
14	(14,15)	2
15	(15,1)	2
1	(1,5)	2
5	(5,13)	6

1+4+1+6+2+10+10 % 16 = **2**
4+1+6+2+10+10 % 16 = **1**
1+6+2+10+10 % 16 = **13**
6+2+10+10 % 16 = **12**
2+10+10 % 16 = **6**
10+10 % 16 = **4**
10 % 16 = **10**

n = number of nodes   **[5]**
m = bytes needed for key   **[4]**
i = i >1 && < m

start: (id + 2^{i-1}) % 2^m
int: (id^i, id^{i+1}) if its last (id^i,id)
succ: id of next node
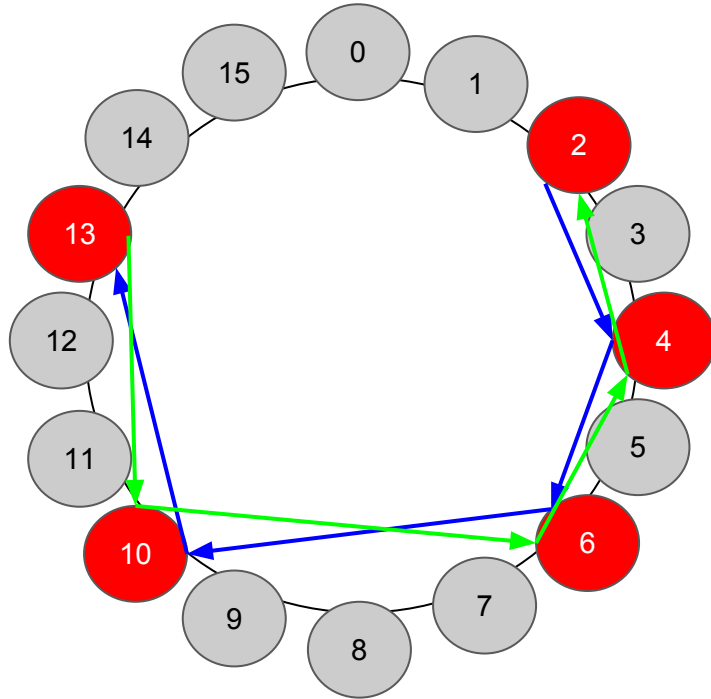
c) min to max

iterative
request
return
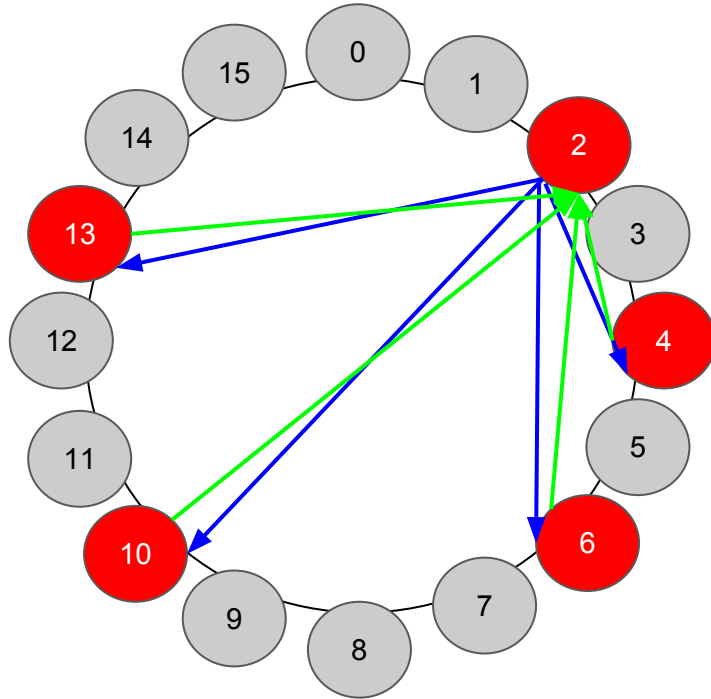
1. N2 asks its successor (N4) for N13
2. N4 returns its successor (N6) to N2
3. N2 asks N6 for N13
4. N6 returns its successor (N10) to N2
5. N2 asks N10 for N13
6. N10 returns its successor (N13) to N2
7. N2 sends a message to N13
8. N13 returns its answer to N2

c) min to max

recursive
request
return

1. N2 sends message to its successor (N4)
2. N4 forwards the message to its successor (N6)
3. N6 forwards the message to its successor (N10)
4. N10 forwards the message to its successor (N13)
5. N13 returns its answer to its predecessor (N10)
6. N10 forwards the answer to its predecessor (N6)
7. N6 forwards the answer to its predecessor (N4)
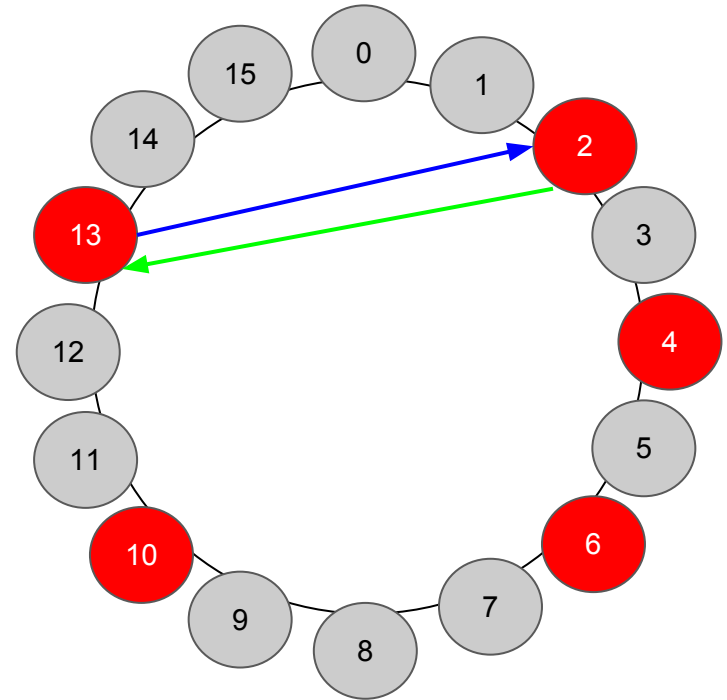8. N4 forwards the answer to its predecessor (N2)

d) max to min

iterative
request
return

1. N13 sends message to its successor (N2)
2. N2 returns its answer to N13

d) max to min

recursive
request
return

1. N13 sends message to its successor (N2)
2. N2 returns its answer to N13