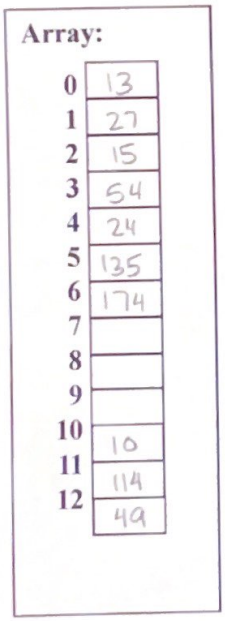
- Using open addressing with linear probing

N = 13



$1 + 1^2 \cdot 1$
$1 + 2^2 \cdot 2$
$1 + 3^2 \cdot 3$

- Using open addressing with quadratic probing

N = 13

Array:

0	13
1	49
2	15
3	54
4	27
5	135
6	174
7	
8	
9	
10	10
11	114
12	24

$54 + 1^2 \bmod 13 = 3$

$114 + 1^2 \bmod 13 = 11$

$49 + 1^2 \bmod 13 = 11$

$49 + 2^2 \bmod 13 = 1$

$174 + 1^2 \bmod 13 =$

$27 + 1^2 \bmod 13 = 2$

$27 + 2^2 \bmod 13 = 5$

$27 + 3^2 \bmod 13 = 10$

$27 + 4^2 \bmod 13 = 4$

$24 + 1^2 \bmod 13 =$

- Using open addressing (division hashing) and the linear-quotient collision path algorithm

N = 13, 4k+3 prime = **19**

LQHashing:

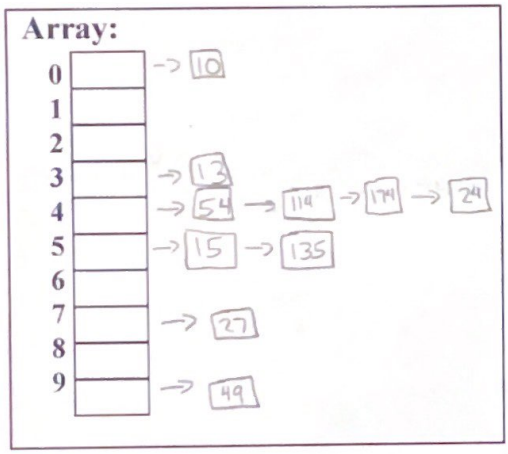
1. i_p = pk % N
2. q=pk/N
   if (q%N != 0)
      offset = q
   else
      offset = 4k+3 prime
3. While collisions:
      $i_{p'}$ = (i_p + offset) % N
4. Set Array[i_p]=key

54/13 = 4
2+4 %13 = 6
49/13 = 3
10+3 %13 = 0
0+3 %13 = 3
174/13 = 13
5+19 %13 = 11
24/13 = 1
11+1 %13 = 12

Array:

0	13
1	27
2	15
3	49
4	
5	135
6	54
7	
8	114
9	
10	10
11	174
12	24

- Bucket hashing where (N=10) and ip = (p_k) % N



15 % 10 = 5
54 % 10 = 4
13 % 10 = 3
10 % 10 = 0
135 % 10 = 5
114 % 10 = 4
49 % 10 = 9
174 % 10 = 4
27 % 10 = 7
24 % 10 = 4

- Come up with your own 15 elements with an array size of 21 using an open addressing algorithm of your choice
- Challenge yourself! Use an algorithm that you are most unfamiliar with to maximize your learning experience

[3, 5, 13, 62, 17, 1, 0, 99, 113, 23, 73, 48, 56, 81, 66]

Elements: 15

mod 21: 3 5 13 10 17 1 0 8 9 10 8 9 4 3 1

N = 21

LQ Hashing From Previous Problem

Array:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	1		3	56	5	81	66	99	113	62	23		13	73	48		17			

23/21 = 1
10+1 mod 21 = 11

73/21 = 3
8+3 mod 21 = 11
11+3 mod 21 = 14

48/21 = 2
9+2 mod 21 = 11
11+2 mod 21 = 13
13+2 mod 21 = 15

81/21 = 3
3+3 mod 21 = 6

66/21 = 3
1+3 mod 21 = 4
4+3 mod 21 = 7