Data Structures & Algorithms

25: Hash Tables

(Collision Resolution)

Dr Ram Prasad Krishnamoorthy

Associate Professor
School of Computing and Data Science

ram.krish@saiuniversity.edu.in

Hash Tables

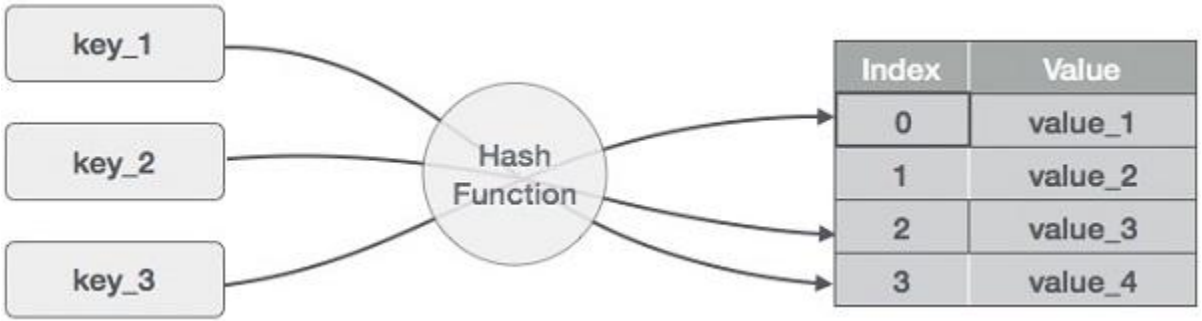
Hash Tables

- A hash table is an effective data structure for implementing dictionaries.
- The built-in dictionaries of Python are implemented with hash tables.
- Indices can be integers, floats, chars, strings.

Hash Tables

Hashing Function

- Function which transforms a key to an index.
- If **h** is a hash function that takes key as a parameter, it will compute index where value is placed.
- It is important to note that hash function returns the same index value every time for same key.



Division Method

Division Method for hashing

- Hashing an integer **k** is to divide **k** by **M** and then to use the remainder **modulo M**.
  - This is called the **division method** of hashing.
  - **h(k) = k mod M**
    - **k** is the key value
    - **M** is the hash table size

Hash Tables

Division Method for hashing

- Hashing an integer **k** is to divide **k** by **M** and then to use the remainder **modulo M**.
  - This is called the **division method** of hashing.
  - **h(k) = k mod M**
    - **k** is the key value
    - **M** is the hash table size

Example:

Hash table size M = 10
Key value, k = 112
Hash function = h(k) = k mod M = 112 mod 10 = 2

Collision

Hash Tables

Collisions

- Collision occurs when one key value stores more than one value. Hash function produces the same position for two values.

- Hash table cannot ensure to have single value corresponding to a key. This is because a hash table has fixed number of keys.

- Any new entry in hash table, after all the slots are filled, causes collision. However, good hash tables use some mechanism to avoid the situations of collisions or to handle them efficiently.

Resolving Collisions

- **Closed addressing** → Chaining
- **Open addressing** → Linear Probing, Quadratic Probing, Double Hashing

Hash Tables

```
code = ['CS', 'CDS', 'MATH', 'STAT']

for elem in code:

 val = 0
 for ch in elem:
   print(f'{ch} --> {ord(ch)}')
   val = val + ord(ch)

 print(f'ASCII sum for {elem} is {val}\n')
```

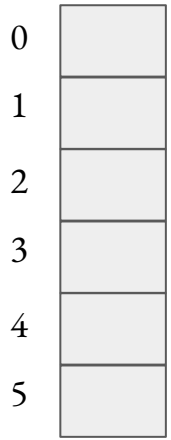
```
C --> 67
S --> 83
ASCII sum for CS is 150

C --> 67
D --> 68
S --> 83
ASCII sum for CDS is 218

M --> 77
A --> 65
T --> 84
H --> 72
ASCII sum for MATH is 298

S --> 83
T --> 84
A --> 65
T --> 84
ASCII sum for STAT is 316
```

Chaining

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

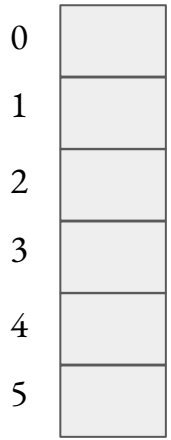


key	index (u)

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

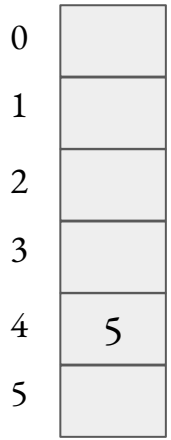


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

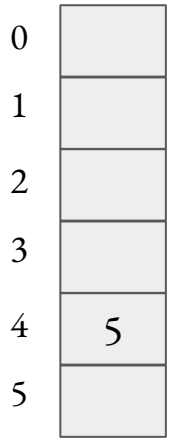


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

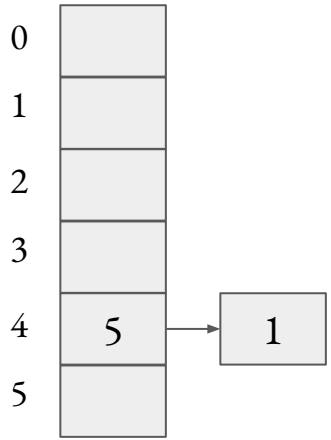


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

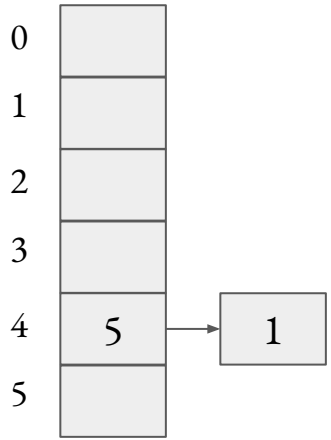


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

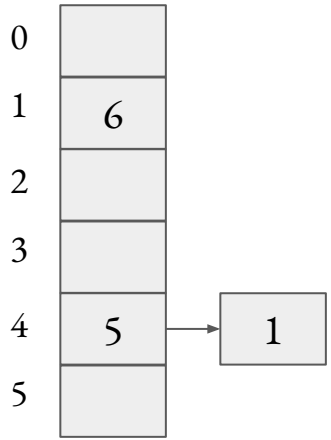


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4
6	(3x6 + 1) % 6 = 19 % 6 = 1

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

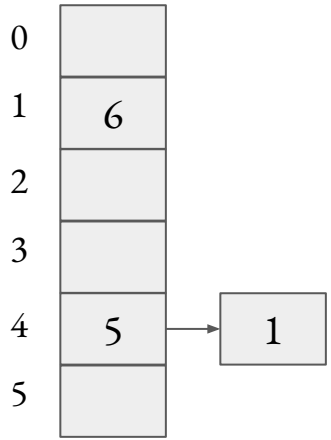


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4
6	(3x6 + 1) % 6 = 19 % 6 = 1

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

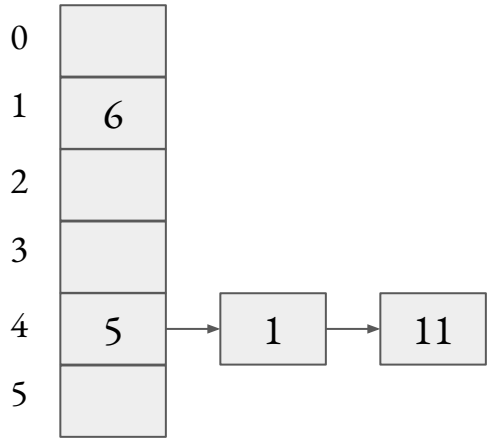


key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4
6	(3x6 + 1) % 6 = 19 % 6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4

Hash Tables

Closed addressing – Chaining

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



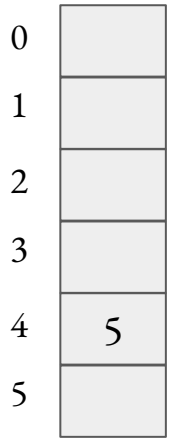
key	index (u)
5	(3x5 + 1) % 6 = 16 % 6 = 4
1	(3x1 + 1) % 6 = 4 % 6 = 4
6	(3x6 + 1) % 6 = 19 % 6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4

Linear Probe

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

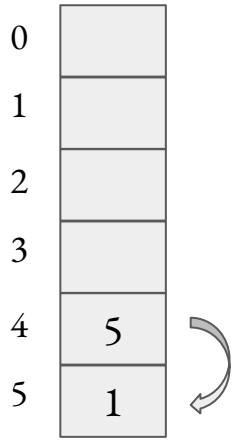


key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

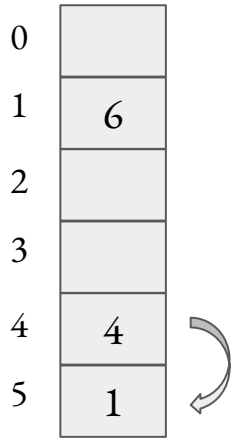


key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

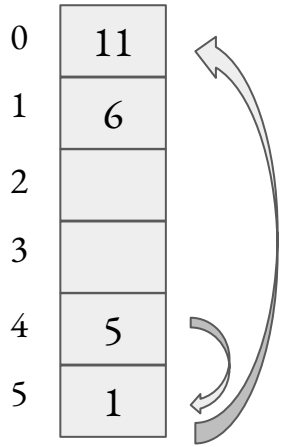


key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

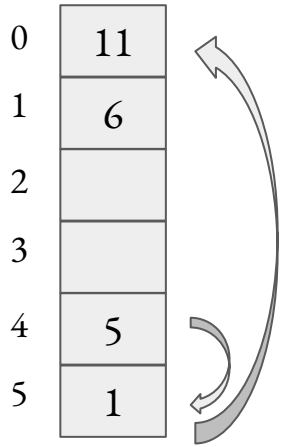


key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	3

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



probing

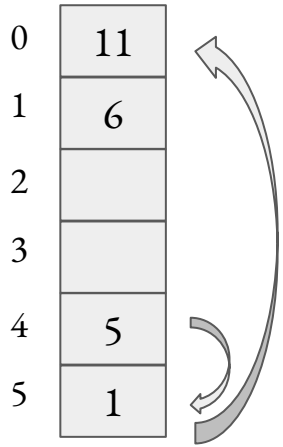
(u + i) % M
i = 0 to M-1

key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	3

Hash Tables

Open addressing – Linear Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



probing

(u + i) % M
i = 0 to M-1

key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	3

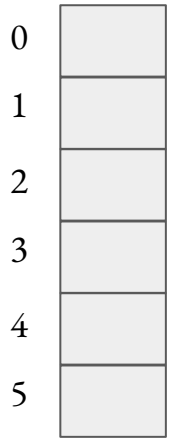
Order of elements in hash table: 11, 6, __, __, 5, 1

Quadratic Probe

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

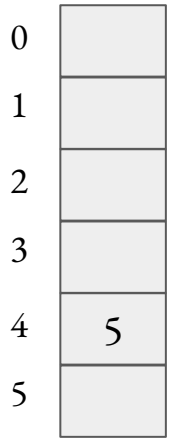
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

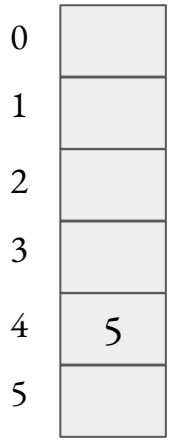
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

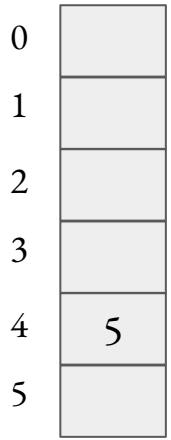
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4

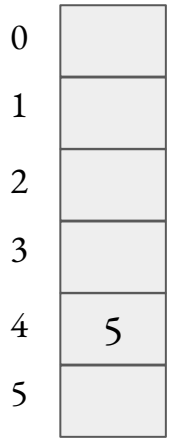
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4
$(4 + 1^2)$ % 6 = 5

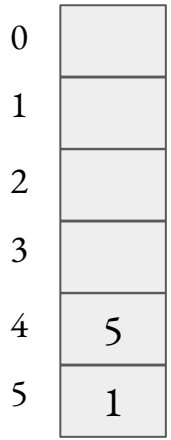
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4
$(4 + 1^2)$ % 6 = 5

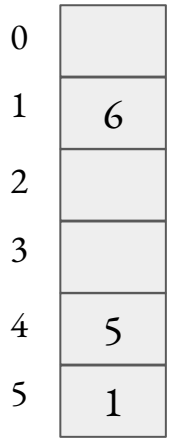
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

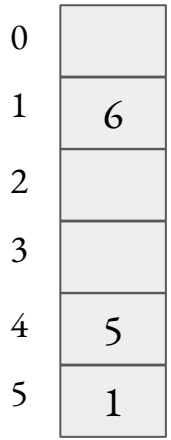
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

0	
1	6
2	
3	
4	5
5	1

$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4

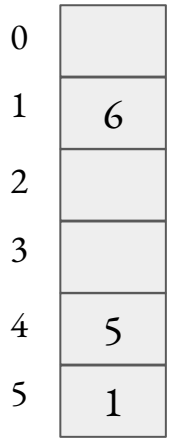
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4
$(4 + 1^2)$ % 6 = 5

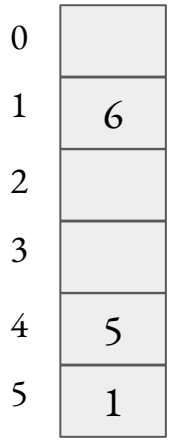
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4
$(4 + 1^2)$ % 6 = 5
$(4 + 2^2)$ % 6 = 2

key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6

0	
1	6
2	11
3	
4	5
5	1

$(u + i^2)$ % M
i = 0 to M-1

$(4 + 0^2)$ % 6 = 4
$(4 + 1^2)$ % 6 = 5
$(4 + 2^2)$ % 6 = 2

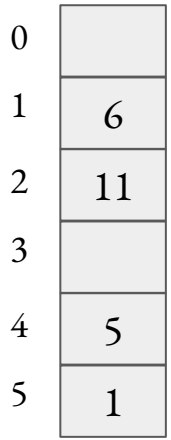
key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	3

Order of elements in hash table:

Hash Tables

Open addressing – Quadratic Probe

keys = 5, 1, 6, 11 h(k) = 3k + 1 M = 6



$(u + i^2)$ % M
i = 0 to M-1

key	index (u)	probe
5	(3x5 + 1) % 6 = 16 % 6 = 4	1
1	(3x1 + 1) % 6 = 4 % 6 = 4	2
6	(3x6 + 1) % 6 = 19 % 6 = 1	1
11	(3x11 + 1) % 6 = 34 % 6 = 4	3

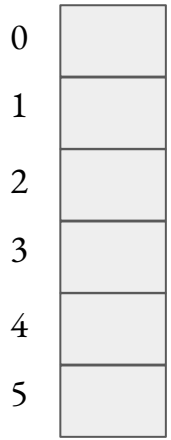
Order of elements in hash table: __, 6, 11, __, 5, 1

Double Hash

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

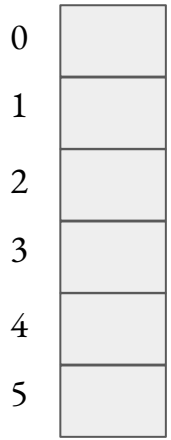
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

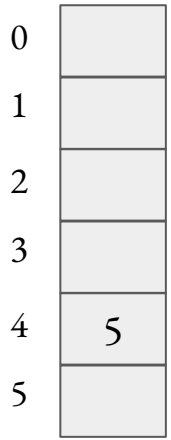
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

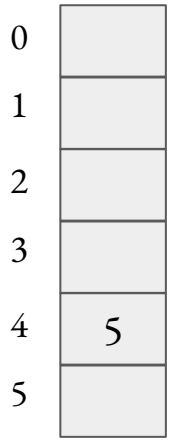
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

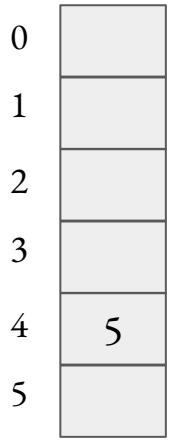
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

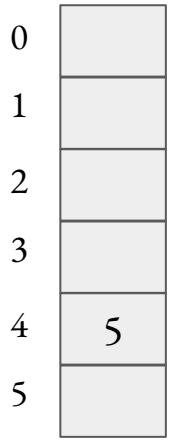
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

(4+3*0)%6 = 4

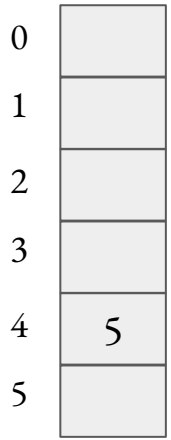
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

(4+3*0)%6 = 4
(4+3*1)%6 = 1

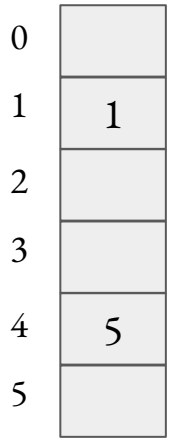
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

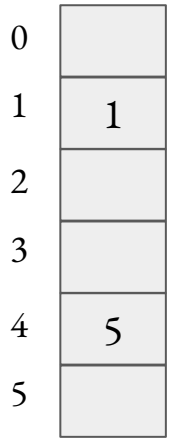
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

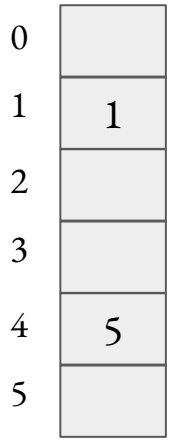
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

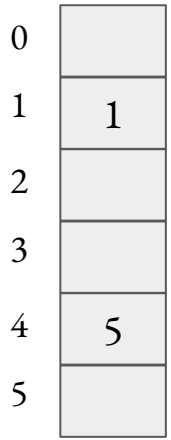
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad M = 6 \quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

(1 + 1*0)%6 = 1
(1 + 1*1)%6 = 2

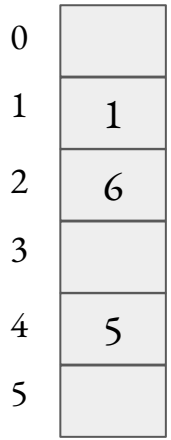
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

(1 + 1*0)%6 = 1
(1 + 1*1)%6 = 2

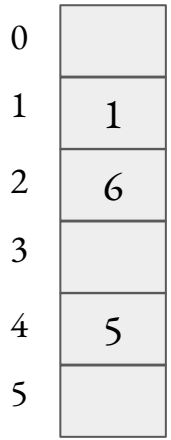
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

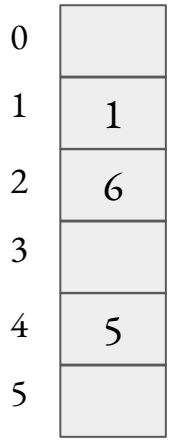
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

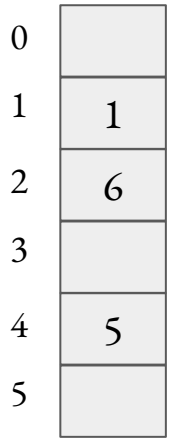
key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	(2x11 + 1)%6 = 5

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$



(u + v*i) % M
i = 0 to M-1

(4 + 5*0)%6 = 4
(4 + 5*1)%6 = 3

key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	(2x11 + 1)%6 = 5

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$

0	
1	1
2	6
3	11
4	5
5	

(u + v*i) % M
i = 0 to M-1

(4 + 5*0)%6 = 4
(4 + 5*1)%6 = 3

key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	(2x11 + 1)%6 = 5

Order of elements in hash table:

Hash Tables

Open addressing – Double Hash

keys = 5, 1, 6, 11 $\quad h_1(k) = 3k + 1 \quad$ M = 6 $\quad h_2(k) = 2k + 1$

0	
1	1
2	6
3	11
4	5
5	

(u + v*i) % M
i = 0 to M-1

key	index (u)	index (v)
5	(3x5 + 1) % 6 = 16 % 6 = 4	
1	(3x1 + 1) % 6 = 4 % 6 = 4	(2x1 + 1)%6 = 3
6	(3x6 + 1) % 6 = 19 % 6 = 1	(2x6 + 1)%6 = 1
11	(3x11 + 1) % 6 = 34 % 6 = 4	(2x11 + 1)%6 = 5

Order of elements in hash table: __, 1, 6, 11, 5, __