Open Addressing
Cryptographic Hashing

...

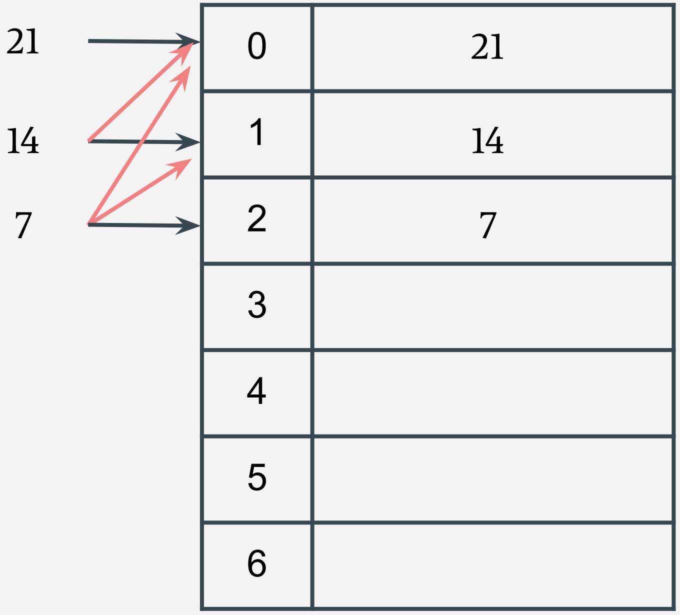
Wonseok

(SWE Intern)

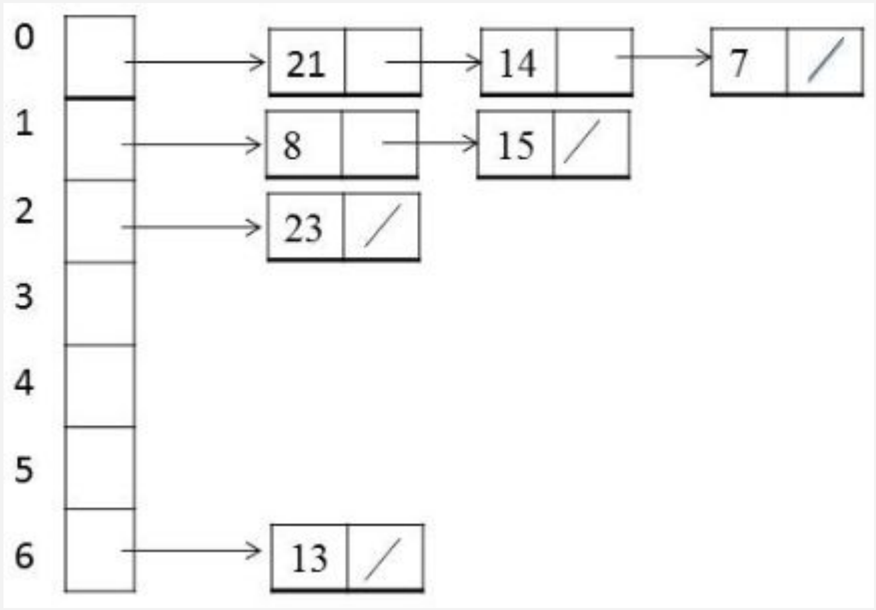
목차

Open Addressing

한 슬롯에 하나의 값만 저장되도록 하는 Array 방식의 해시.
해시값 충돌 시 다른 주소에 저장하므로 Open Addressing이라 함.



< Open Addressing >



< Chaining >

Assumption

Input: [21, 14, 7, 23, 13, ...] -> **n** (Elems)

0	21
1	14
2	7
3	
4	
5	
6	

m (Slots)

- Each slot in the table can store one and only one entry
- Slot size **m** is less than or equal to elem size **n**
- When **U** is universe of keys and **T** is trial count, hash function **h(U, T)** satisfies:

$$h: U \times \{0, 1, 2, \ldots, m-1\} \rightarrow \{0, 1, 2, \ldots, m-1\}$$

Open Addressing Needs **Probing** !

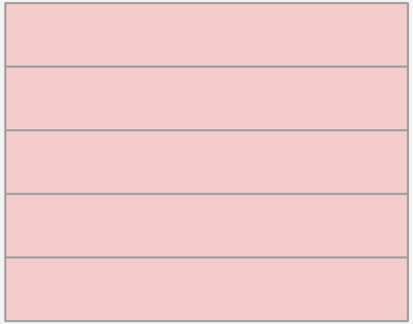
Probing

How will we resolve collisions and find an empty slot?

Uniform Hashing Assumption

Each key is equally likely to have one of m! Permutation on its probe sequence

(O)



h(k, 0...4) = {0, 1, 2, 3, 4}

(X)

h(k, 0...4) = {0, 2, 4}

Linear Probing | Double Hashing

0	
1	1
2	6
3	11
4	
5	

If there's a collision, assign the value to the next free space

$$\mathbf{h(k, i) = h'(k) + i}$$

ex) h'(k) = k mod 6

But Linear Probing can result in **clustering** ,

which slows down hash's performance.

Linear Probing | **Double Hashing**

Use two hash functions to resolve clustering

$$h(k, i) = [\, h_1(k) + h_2(k) * i \,] \bmod m$$

ex) $h_1(k) = k \bmod 7$, $h_2(k) = k \bmod 11$

$m = 6$

0	71	$h(71, \mathbf{1}) = (1 + 5 * 1) \bmod 6 = 0$	$h_1(71) = 1$, $h_2(71) = 5$
1	1	$h(1, \mathbf{0}) = (1 + 1 * 0) \bmod 6 = 1$	$h_1(1) = 1$, $h_2(1) = 1$
2			
3			
4			
5	15	$h(15, \mathbf{1}) = (1 + 4 * 1) \bmod 6 = 5$	$h_1(15) = 1$, $h_2(15) = 4$

We still need modification over those hash functions.
Try 77, 154, ... and 11, 22, 33, ... it will always return the same value.

Linear Probing | **Double Hashing**

$$h(k, i) = \{ h_1(k) + h_2(k) * i \} \bmod m$$

$$h_1(k) = k \bmod 7,\ h_2(k) = k \bmod 11,\ m = 6$$

Observation

- Once k is decided, $h_1(k)$ and $h_2(k)$ are constants, only i is variable.
- A cycle in hash values happens with period in relation to GCD between m and $h_2(k)$.

(1) when k is 14, h(14) = 0, h(14) = 3, **h(14, i) = (0 + 3*i) mod 6 = [3, 0,** 3, 0, ...**]**
(2) when k is 13, h(13) = 6, h(13) = 2, **h(13, i) = (6 + 2*i) mod 6 = [0, 2, 4,** 0, 2, 0, ...**]**

$$\text{Length of cycle is } \frac{m}{GCD(m, h_2(k))}$$

Solution

Ensure that GCD between m and h(k) is 1, so that the length of cycle is m (size of the array)

$$\boxed{m = 2^x,\ h_2(k) \text{ is odd}}$$

Insert, Search, Delete

0	200
1	300
2	400
3	
4	404
5	deleted
6	604
7	
8	108
9	

< Linear Probing >

Insert(k) : Keep probing until an empty slot is found.

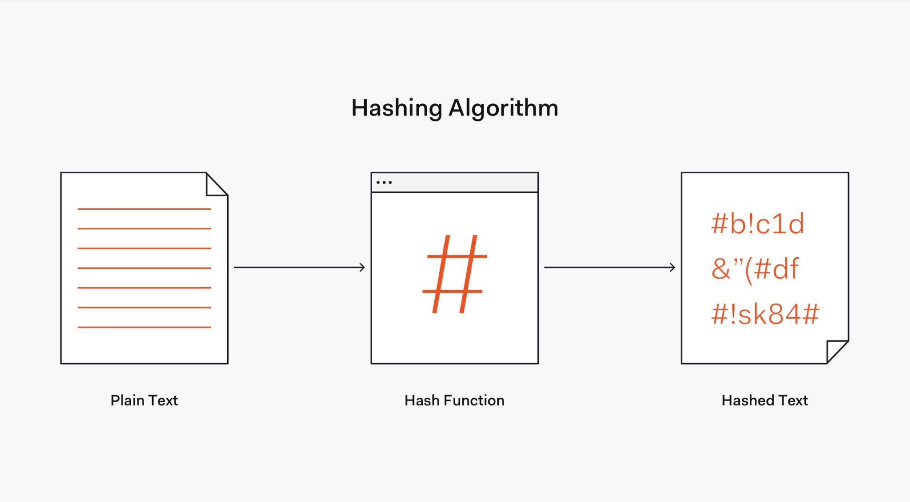
Table Doubling when load factor is > 0.5

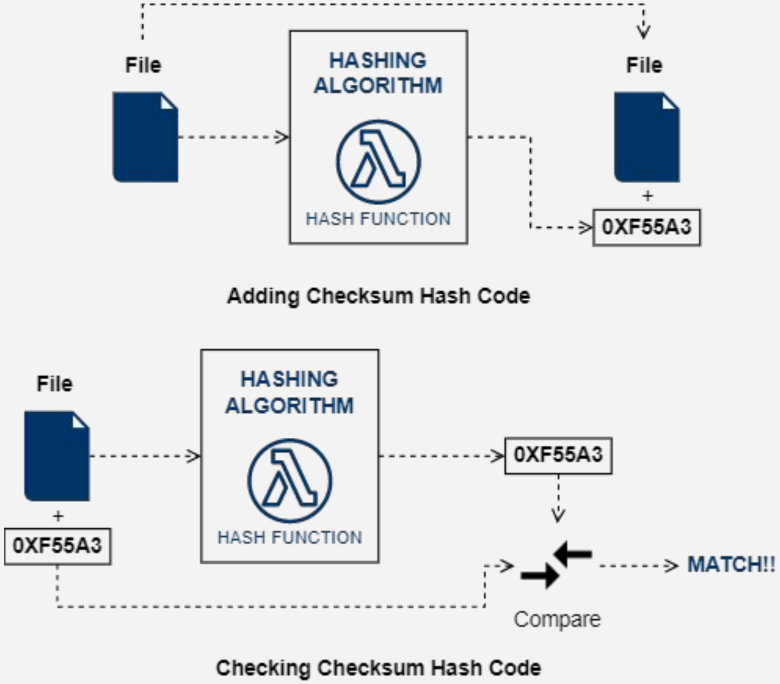
Search(k) = Probing (Linear Probing/Double Hashing)

Delete(k) : Delete and mark it with "deleted" flag

- Free slot for insertion
- Occupied slot for search

Examples





< Password Storage >

< File Checksum >