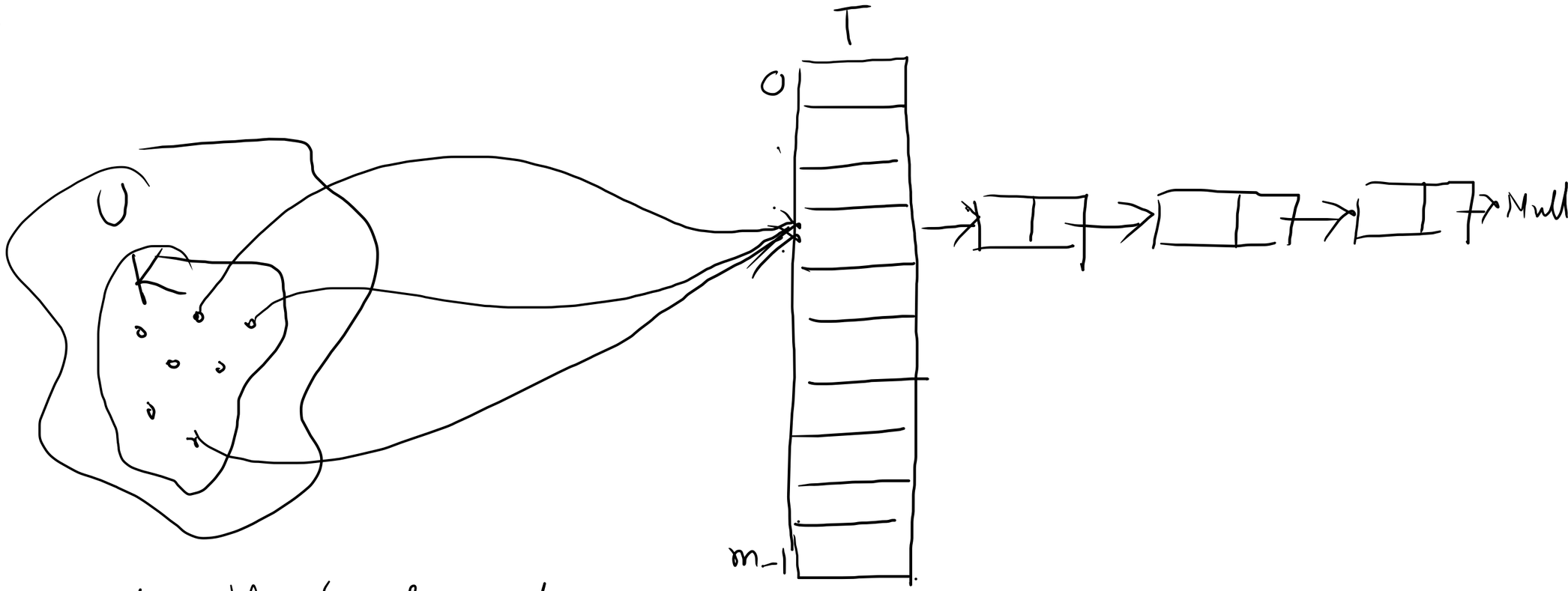
$m \ll n$

collision occurs ⇒

Resolve the collision

- collision resolution by chaining ✓
- open addressing.

# Chaining

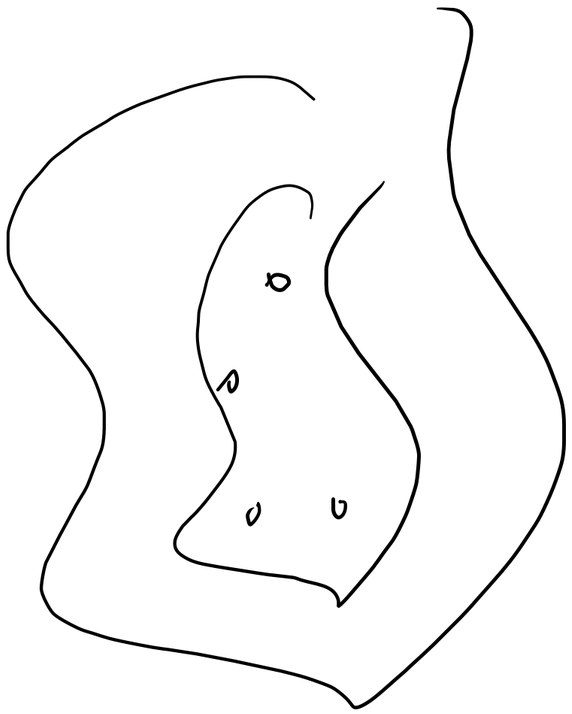


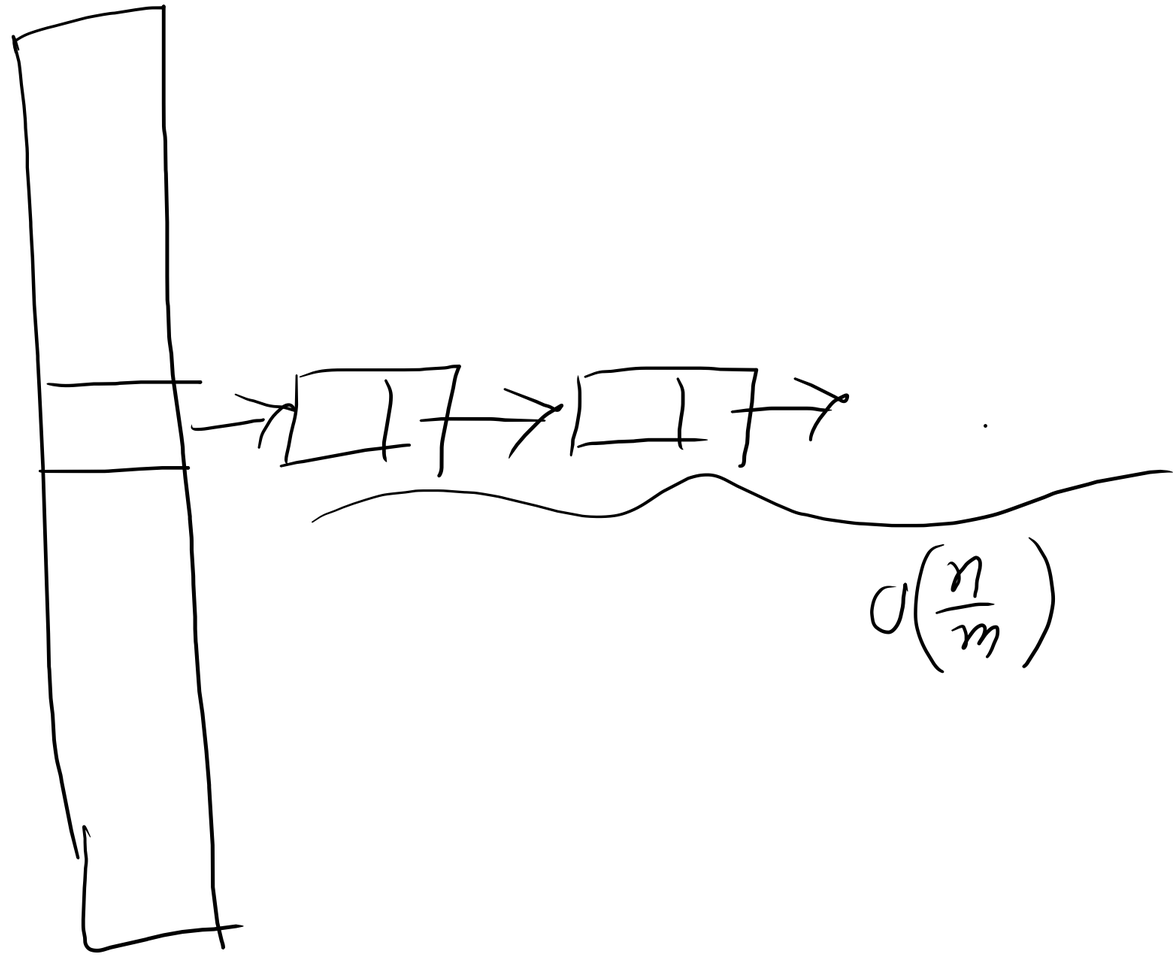

Insert: compute the hash value

similar { Delete / search } then traverse the list of the index of the hash value and insert in appropriate place.

worst case: $O(n)$

$\frac{n}{m} = \alpha$






$O\left(\frac{n}{1+\alpha}\right)$

THIS become constant when $\alpha = \text{const}$. $m = \Omega(n)$

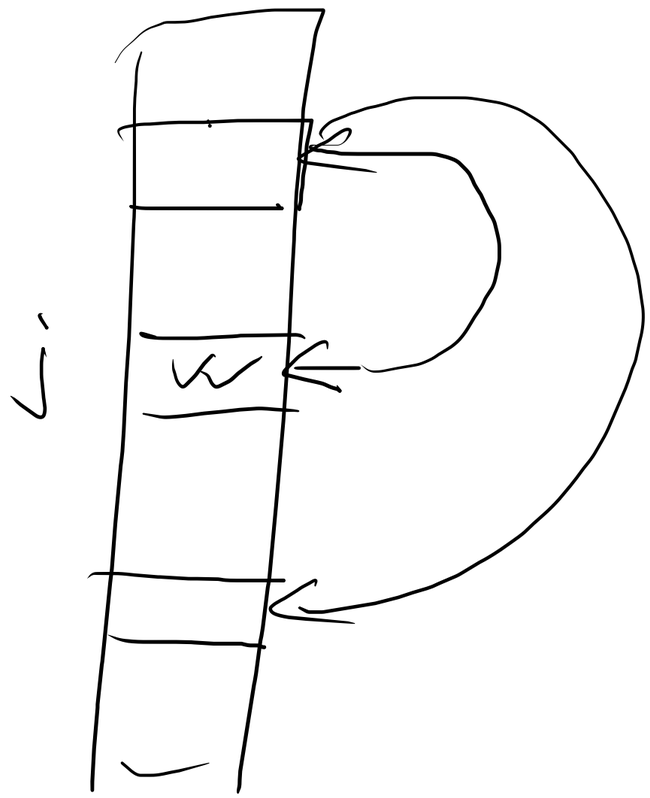
## Open addressing

- It is a collision resolution technique
- no chaining instead all items/keys are stored in the table itself.
- $m \geq n$

Probe - try

h

$k_1\ k_2 \cdots k_n$.




$h(k_i) = j$

integer.

$l$ = value

$(h(k_i) + l)$

## Probing

- Hash function specifies order of slots to probe for a key not just a single slot.
-

Hash function

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

$$\Rightarrow h(k, i) \rightarrow j$$

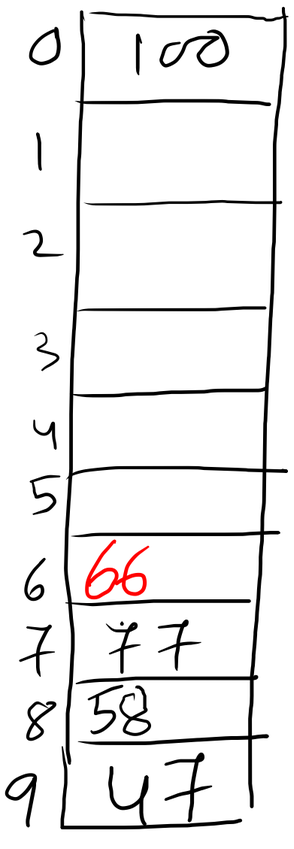
$$\langle h(k,0), h(k,1), h(k,2), h(k,3), \ldots, h(k, m-1) \rangle$$

Given a hash function $h(k, 0)$ start accessing at index $h(k)$. If collision occurs check the next slot at position $h(k, 1)$, ... and continue.

# Linear probing

$h(K, i) = (h_1(k) + i) \bmod m$ , $h_1(k) = k \bmod 10$

insert, 66, 77, 58, 100, 47




$h(66, 0) = 6$ ←

$h(77, 0) = 7$ ←

$h(58, 0) = 8$ ←

$h(100, 0) = 0$ ←

$h(47, 0) = 7$ collision

$h(47, 1) = 8$ collision

$h(47, 2) = 9$ free

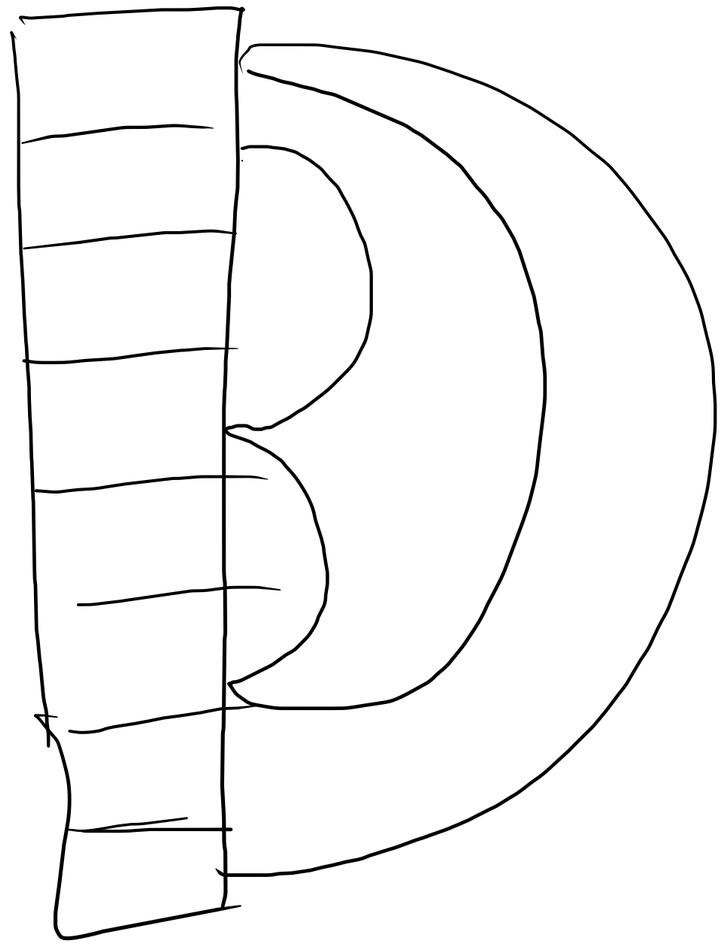
Drawback

Primary clustering

Quadratic Probing

$$h(k,i) = (h_1(k) + i^2 + i + 1) \bmod m$$

47, 57, 37, 27, 17, 7,



Drawback

Secondary clustering

# Double hashing

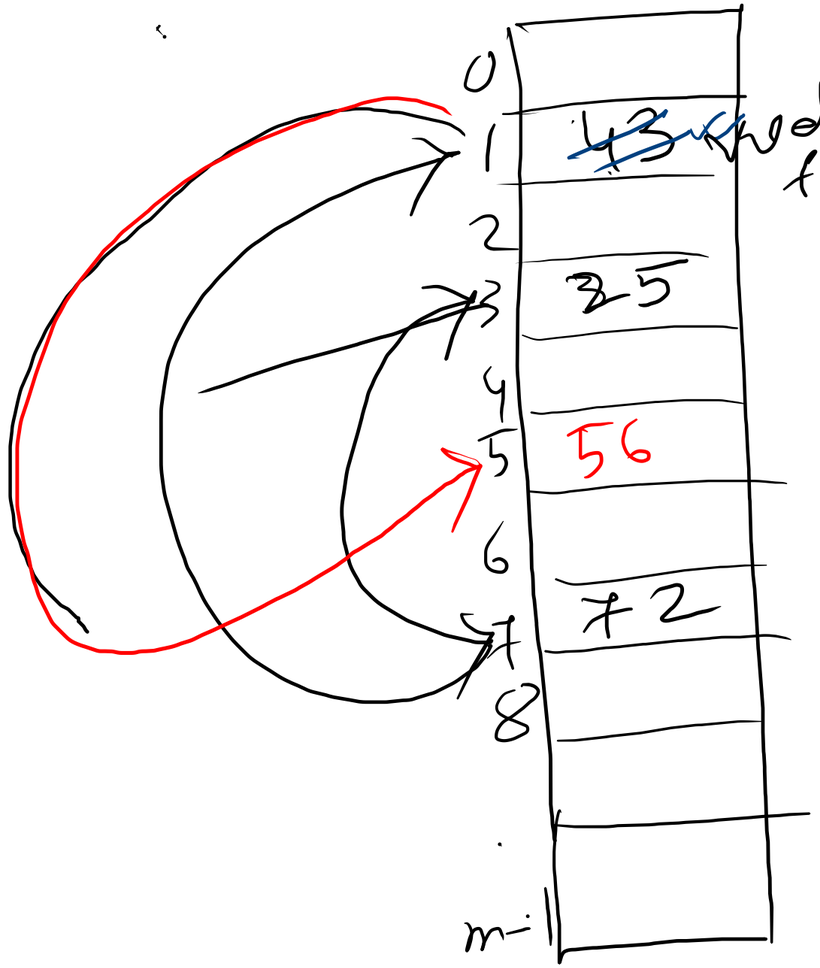
$$h(k,i) = \left(h_1(k) + i h_2(k)\right) \bmod m .$$

H.W $h(k,i) = (h_1(k) + i\, h_2(k)) \bmod m$ .

- $h_1(k) = k \bmod 13$

  $h_2(k) = 1 + k \bmod 11$

insert 79, 69, 72, 50, 96, 14,



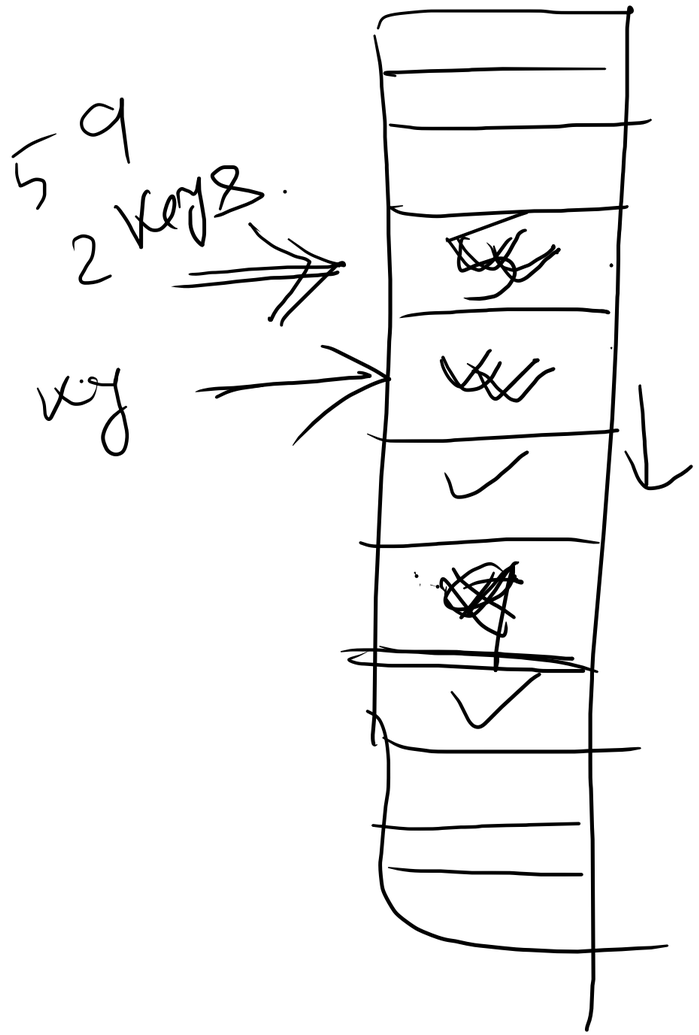

$h(56, 0) = 3$ ✓

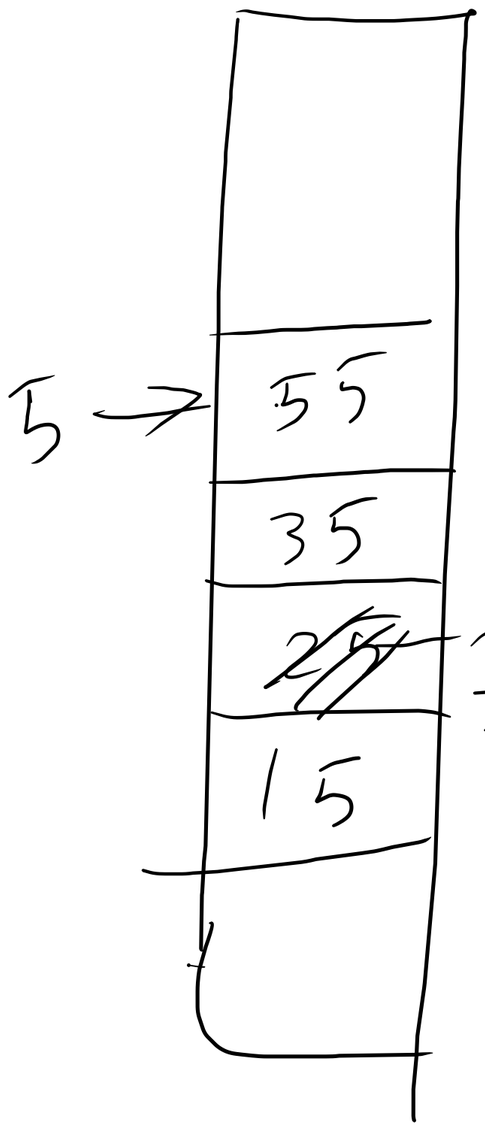
$h(56, 1) = 7$

$h(56, 2) = 1$

$h(56, 3) = 5$.

delete 43 before inserting 56.





55, 35, 25, 15

delete 25

search 15