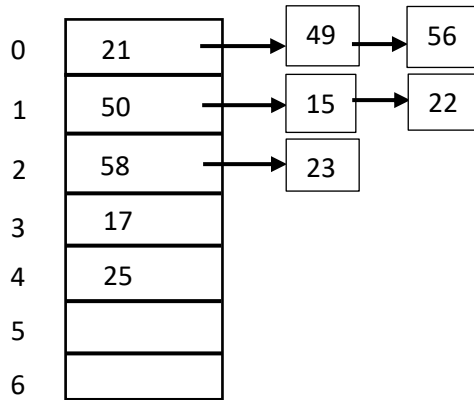


Chaining

$$\text{hash}(\text{key}) = \text{key} \% 7$$

keys = {50, 21, 58, 17, 15, 49, 56, 22, 23, 25}



Hash Table (Array of Linked List Headers)

Performance

m = No of slots in Hash Table

n = No of keys to be inserted

load factor $\alpha = n/m$

Expected chain length = α

Expected time to search = $O(1 + \alpha)$

Expected time to Insert/Delete = $O(1 + \alpha)$

Data Structure for Strong Chains

- ⇒ Linked List (search $O(I)$, delete $O(I)$, insert $O(I)$)
- ⇒ Dynamic Sized Arrays (vector in c++, ArrayList in java, list in python)
- ⇒ Self-Balancing BST (AVL Tree, Red Black Tree) ($O(\log I)$)