

# ECAP770

## ADVANCE DATA STRUCTURES

---

Ashwani Kumar  
Assistant Professor



# Learning Outcomes



After this lecture, you will be able to

- understand rehashing,
- load factor.

# Rehashing

- It is process of re-calculating the hash code of already stored entries.
- The Hash table provides Constant time complexity of insertion and searching, provided the hash function is able to distribute the input load evenly.
- In case of Collision, the time complexity can go up to  $O(N)$  in the worst case

# Rehashing

- Rehashing of a hash map is done when the number of elements in the map reaches the maximum threshold value.

# Rehashing

- When load factor increases to more than its predefined value, complexity increases.
- To overcome this problem, size of array is increased, and all the values are hashed again and stored in new double size array to maintain a low **load factor** and **complexity**.

# Load factor

- Load factor is number of element (n) divide by number of bucket (m).
- Load factor ( $\lambda$ ) = $n/m$
- $\lambda < 1$  i.e.  $m>n$
- if  $\lambda < 1$  then no need to apply rehashing
- if  $\lambda > 1$  then we need to increase number of buckets
- Increase in bucket size is known as rehashing.

# Rehashing steps

-Increase number of buckets.

-Modify hash function

- Hash function before rehashing :  $x \bmod m$

after rehashing  $x \bmod m'$

-apply changed hash function to existing elements.

# m' calculation

- $m' = \text{closest prime number of } 2m$
- Example:
- $m=3 \quad m'= 2(3) = 6$
- Closet prime number = 5 or 7.

# Example: Rehashing

$$h(x) = x \bmod 3$$

Elements: 12, 13, 14



$$n = 3$$

$$m = 3$$

$$\text{Load factor } (\lambda) = 1$$

$\lambda$  is not less than 1 so need to perform rehashing

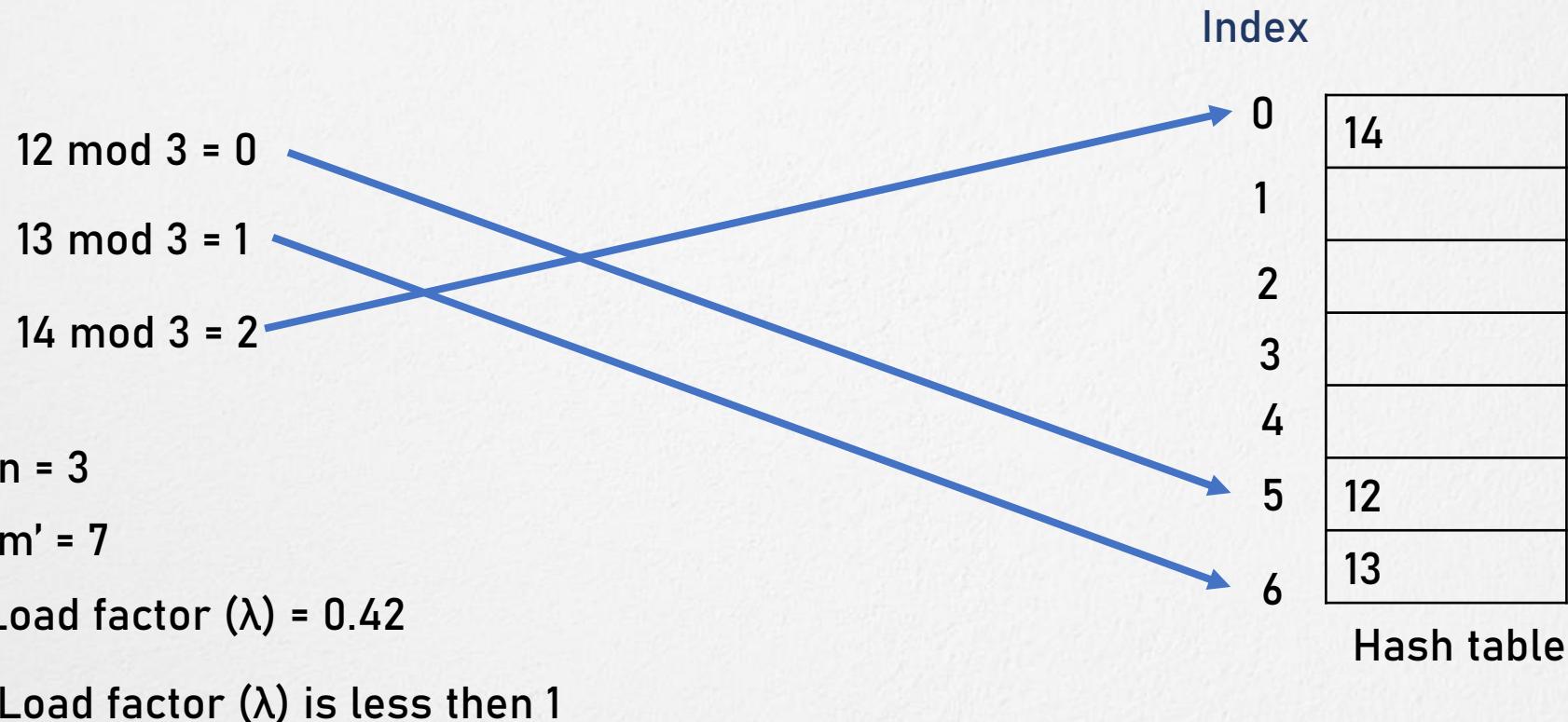
# Example: Rehashing

- $m' = 2n$
- $m' = 2 \times 3 \Rightarrow 6$
- Nearest prime numbers are 5 and 7
- $m' = 7$
- $x \bmod m' \Rightarrow x \bmod 7$

# Example: Rehashing

$$h(x) = x \bmod 7$$

Elements: 12, 13, 14



That's all for now...