

# Hashing

Dr. Abu Nowshed Chy

Department of Computer Science and Engineering  
University of Chittagong

nowshed@cu.ac.bd 01764207358

Faculty Profile

# Hashing

---

Hash tables are used for keeping values with a key.



Just imagine a locker. You can only open them if you have the keys.



# Hashing

---

Assume you have a hash table named **Users** wherein username is used as key and the value is the name. It will be like this:

*1st record in Hash Table*

Key: jsmith

Value: John Smith

*2nd record in Hash Table*

Key: jdoe

Value: Jane Doe





# Properties of Hash Function

---

## Property 1: Deterministic

No matter how many times you parse through a particular input through a hash function you will always get the same result.

## Property 2: Quick Computation

The hash function should be capable of returning the hash of an input quickly. If the process isn't fast enough then the system simply won't be efficient.





# Properties of Hash Function

---

## Property 3: Pre-Image Resistance

What pre-image resistance states is that given  $H(A)$  it is infeasible to determine  $A$ , where  $A$  is the input and  $H(A)$  is the output hash.

## Property 4: Small Changes In Input Changes the Hash

Even if you make a small change in your input, the changes that will be reflected in the hash will be huge.



# Properties of Hash Function

---

## Property 5: Collision Resistant

Given two different inputs A and B where  $H(A)$  and  $H(B)$  are their respective hashes, it is infeasible for  $H(A)$  to be equal to  $H(B)$ .

## Property 6: Puzzle Friendly

It should be difficult to select an input that provides a pre-defined output. Thus, the input should be selected from a distribution that's as wide as possible.





# Hashing

---

Consider inserting the keys

10, 22, 31, 4, 15, 28, 17, 88, and 59

into a hash table of length  $m=11$  using open addressing with the primary hash function  $h(k) = k \bmod m$ . Illustrate the result of inserting these keys using collision avoidance through linear probing. What is the resultant hash table?





# Hashing

---

0	22
1	88
2	
3	
4	4
5	15
6	28
7	17
8	59
9	31
10	10





## Consider inserting the keys

12, 18, 13, 2, 3, 23, 5 and 15

into a hash table of length  $m=10$  using open addressing with the primary hash function  $h(k) = k \bmod m$ . Illustrate the result of inserting these keys using collision avoidance through linear probing.

# Hashing

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(A)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(C)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

(D)

