SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : **19CS307**- DATA STRUCTURES

II YEAR / III SEMESTER

Unit V- **SORTING AND SEARCHING**

Topic : **Hashing and Hash Functions**

Hashing and Hash Functions

Hashing

- Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.

- (OR)

- In CS, a **hash table**, or a **hash map**, is a data structure that associates keys (names) with values (attributes).
  - Look-Up Table
  - Dictionary
  - Cache
  - Extended Array

Cont..

- In hashing, large keys are converted into small keys by using **hash functions**.
- The values are then stored in a data structure called **hash table**.
- The idea of hashing is to distribute entries (key/value pairs) uniformly across an array.
- Each element is assigned a key (converted key).

Example

- Some examples of how hashing is used in our lives include:
- In universities, each student is assigned a unique roll number that can be used to retrieve information about them.
- In libraries, each book is assigned a unique number that can be used to determine information about the book, such as its exact position in the library or the users it has been issued to etc.

Hashing is implemented in two steps

- An element is converted into an integer by using a hash function.
- This element can be used as an index to store the original element, which falls into the hash table.
- The element is stored in the hash table where it can be quickly retrieved using hashed key.
- hash = hashfunc(key)
  index = hash % array_size

Hash Function

- A hash function is any function that can be used to map a data set of an arbitrary size to a data set of a fixed size, which falls into the hash table.
- The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes.
- Advantage
- Easy to compute: It should be easy to compute and must not become an algorithm in itself
- Uniform distribution: It should provide a uniform distribution across the hash table and should not result in clustering.
- Less collisions: Collisions occur when pairs of elements are mapped to the same hash value. These should be avoided.

Some hash functions

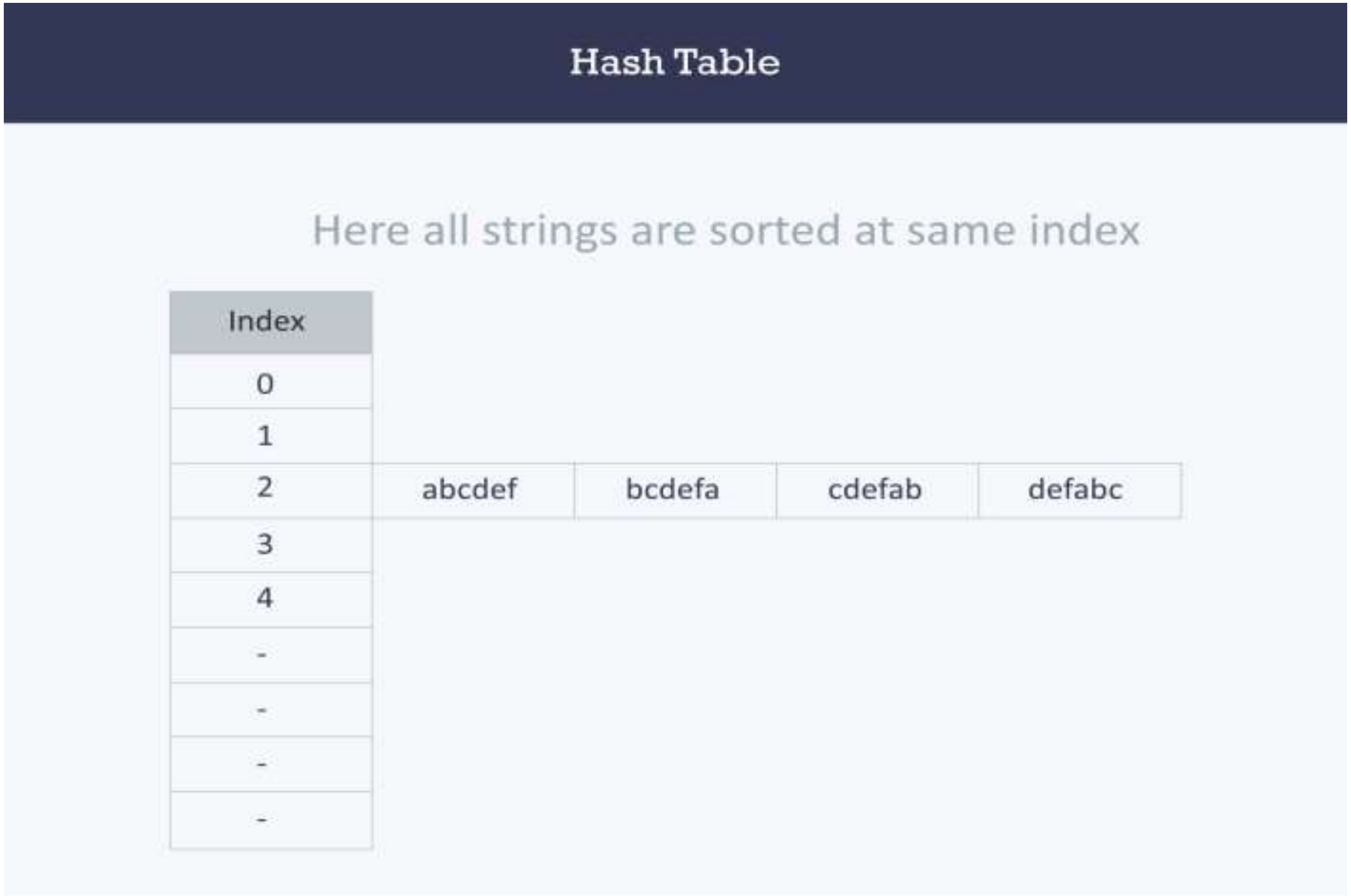
- Middle of square
  - H(x):= return middle digits of x^2
- Division
  - H(x):= return x % k
- Multiplicative:
  - H(x):= return the first few digits of the fractional part of x*k, where k is a fraction.

Need for a good hash function

- Assume that you have to store strings in the hash table by using the hashing technique {“abcdef”, “bcdefa”, “cdefab” , “defabc” }.
- The ASCII values of a, b, c, d, e, and f are 97, 98, 99, 100, 101, and 102 respectively.
- Since all the strings contain the same characters with different permutations, the sum will 599.
- The hash function will compute the same index for all the strings and the strings will be stored in the hash table in the following format.

Example

➢Assume that you have to store strings in the hash table by using the hashing technique {“abcdef”, “bcdefa”, “cdefab” , “defabc” }.



Hash Table

Here all strings are sorted at same index

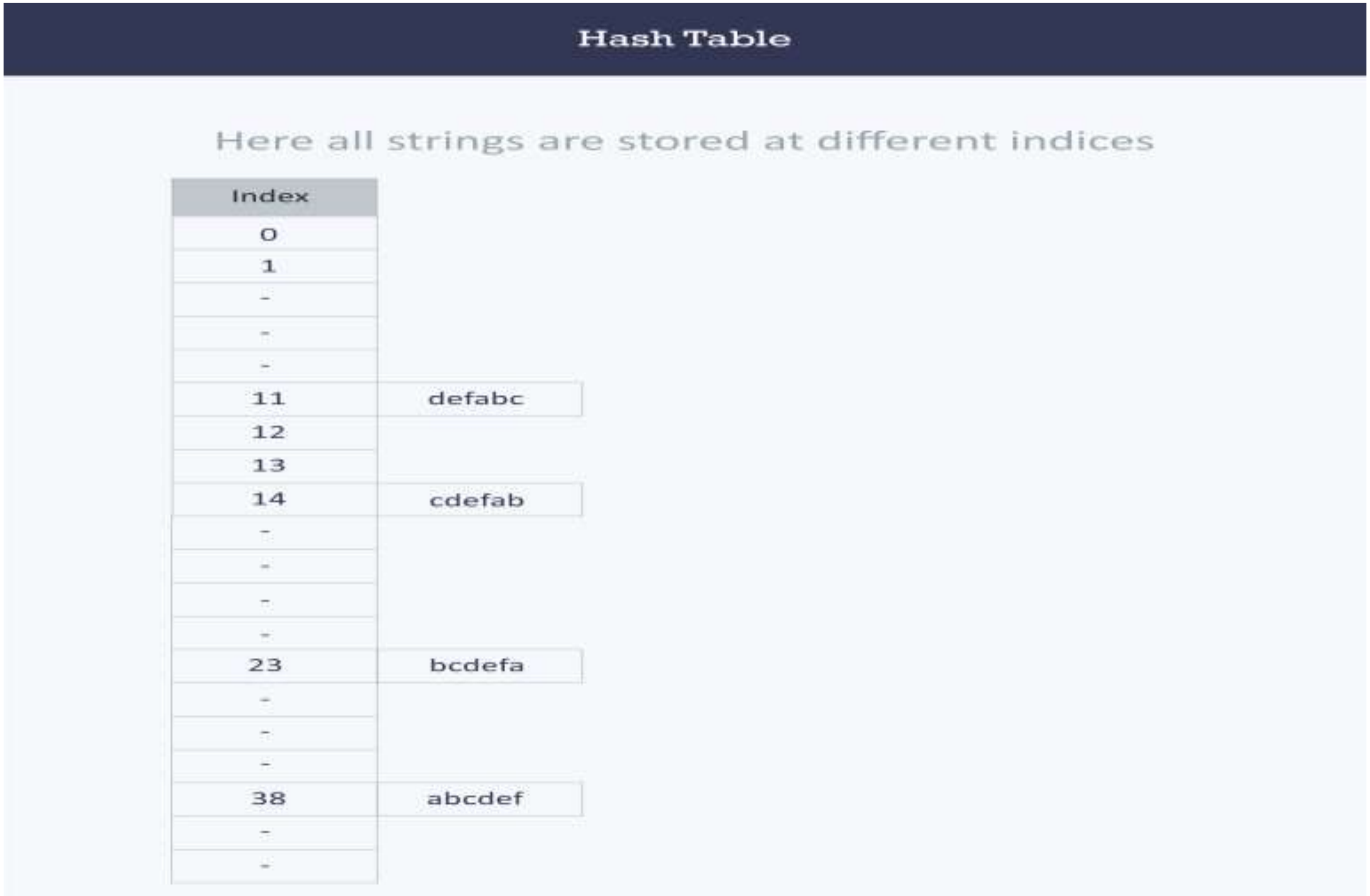
Index				
0				
1				
2	abcdef	bcdefa	cdefab	defabc
3				
4				
-				
-				
-				
-				

Cont..

- Let’s try a different hash function. The index for a specific string will be equal to sum of ASCII values of characters multiplied by their respective order in the string after which it is modulo with 2069 (prime number).

String	Hash function	Index
abcdef	(971 + 982 + 993 + 1004 + 1015 + 1026)%2069	38
bcdefa	(981 + 992 + 1003 + 1014 + 1025 + 976)%2069	23
cdefab	(991 + 1002 + 1013 + 1024 + 975 + 986)%2069	14
defabc	(1001 + 1012 + 1023 + 974 + 985 + 996)%2069	11

Cont..



Hash Table

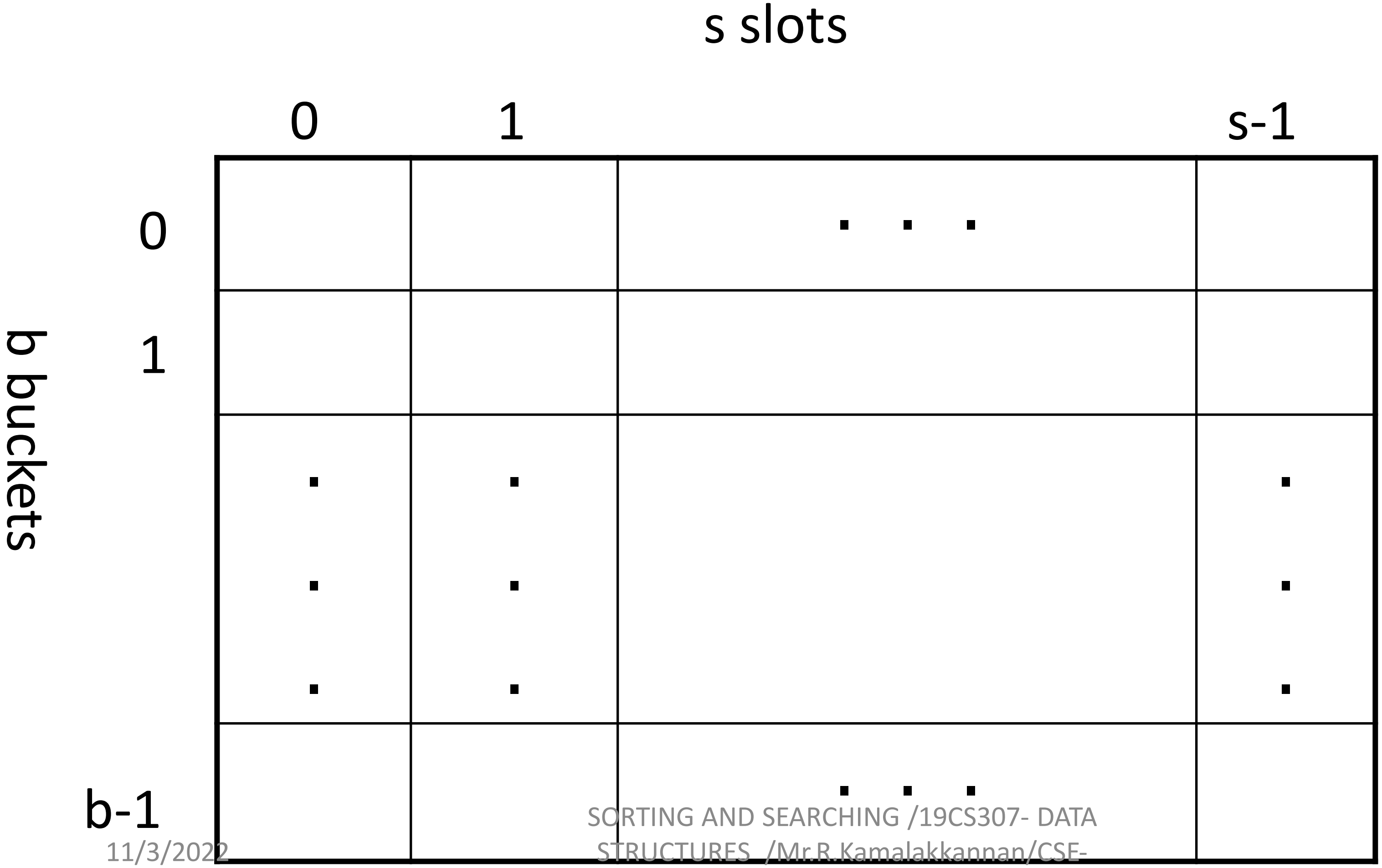
Here all strings are stored at different indices

Index	
0	
1	
-	
-	
-	
11	defabc
12	
13	
14	cdefab
-	
-	
-	
-	
23	bcdefa
-	
-	
-	
38	abcdef
-	
-	

Hash table

- A hash table is a data structure that is used to store keys/value pairs.
- It uses a hash function to compute an index into an array in which an element will be inserted or searched.
- By using a good hash function, hashing can work well.

Hash table



Syntax

- void countFre(string S)
- {
- for(char c = ‘a’;c <= ‘z’;++c)
- {
- int frequency = 0;
- for(int i = 0;i < S.length();++i)
- if(S[i] == c) frequency++;
- cout << c << ‘ ‘ << frequency << endl;
- } }

Search vs. Hashing

- Search tree methods: key comparisons
  - – Time complexity: O(size) or O(log n)
- Hashing methods: hash functions
  - – Expected time: O(1)
- Types
  - – Static hashing (section 8.2)
  - – Dynamic hashing (section 8.3)

Thank you