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

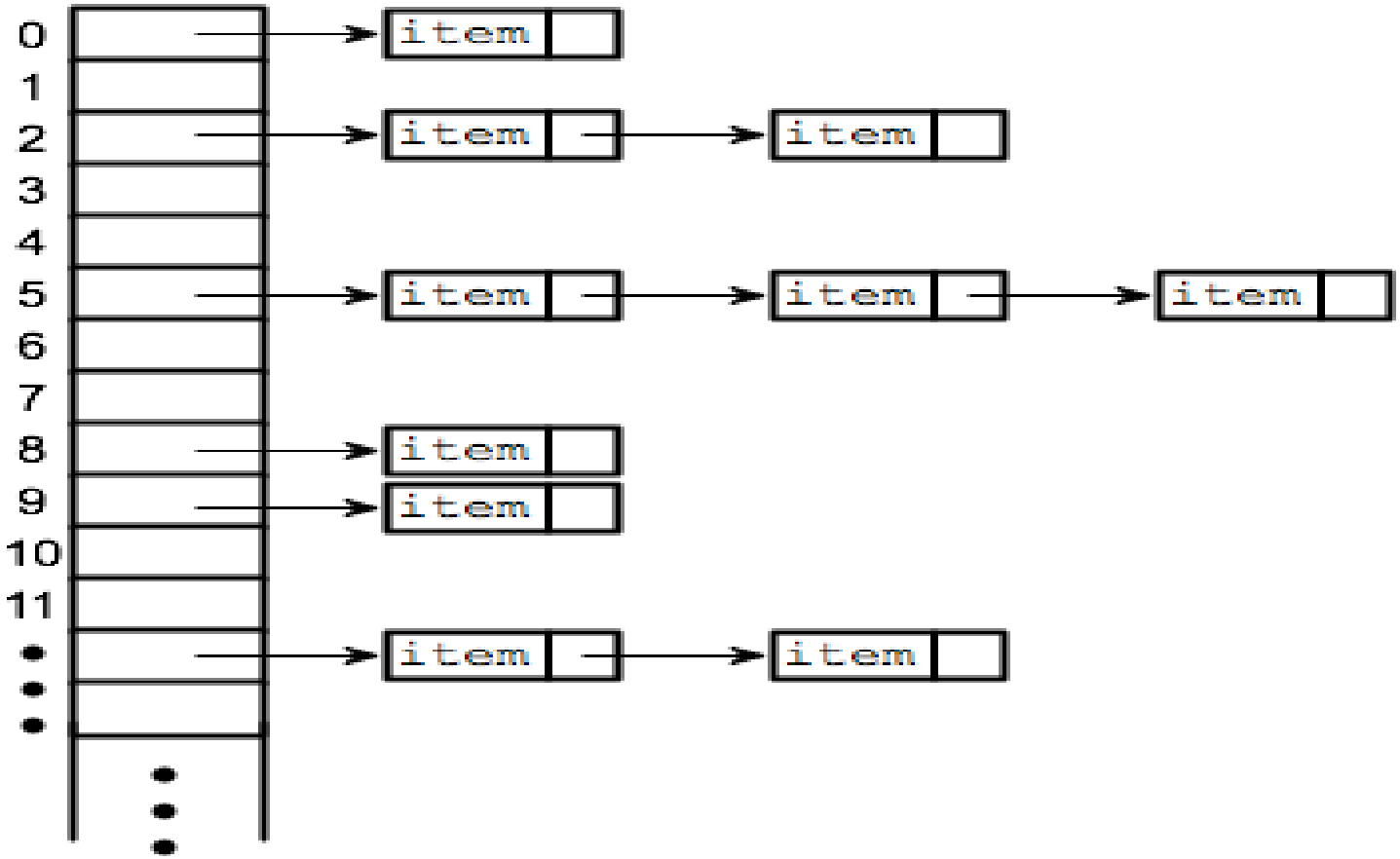
Unit V- **SORTING AND SEARCHING**

TOPIC: Separate Chaining *(open hashing)*

Separate Chaining *(open hashing)*

- Separate chaining is one of the most commonly used collision resolution techniques.
- It is usually implemented using linked lists. In separate chaining, each element of the hash table is a linked list.
- To store an element in the hash table you must insert it into a specific linked list.
- If there is any collision (i.e. two different elements have same hash value) then store both the elements in the same linked list.

Separate chaining



Cont..

- In the following image, **CodeMonk** and **Hashing** both hash to the value **2**.
- The linked list at the index **2** can hold only one entry, therefore, the next entry (in this case **Hashing**) is linked (attached) to the entry of **CodeMonk**.

Implementation of hash tables with separate chaining (open hashing)

- Hash function will return an integer from 0 to 19.
- Syntax
- vector <string> hashTable[20];
- int hashTableSize=20;

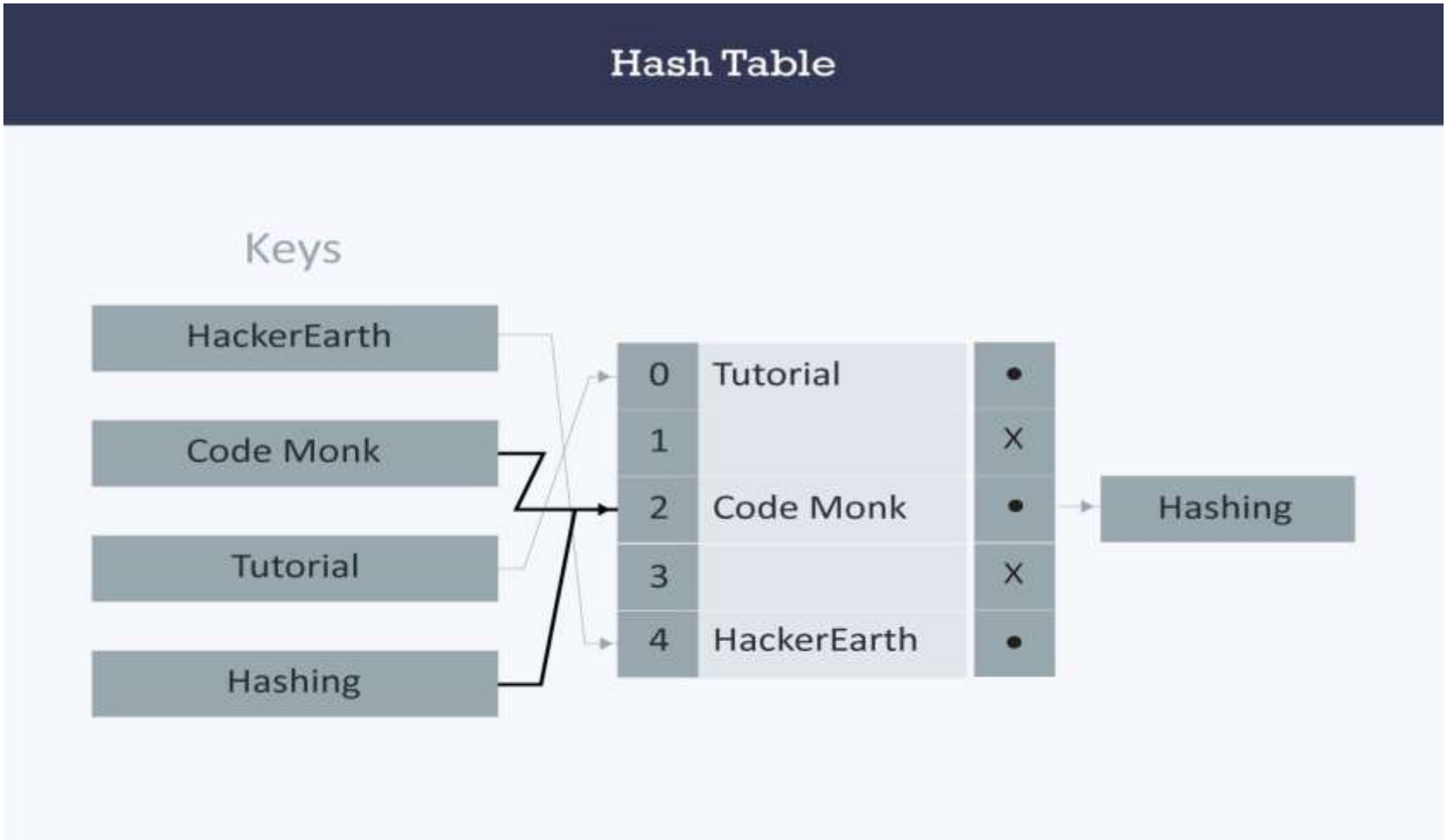
Insert Syntax

- void insert(string s)
- { // Compute the index using Hash Function
  int index = hashFunc(s);
- // Insert the element in the linked list at the particular index
  hashTable[index].push_back(s);
- }

Search Syntax

- void search(string s)
- {
- //Compute the index by using the hash function int index = hashFunc(s);
- //Search the linked list at that specific index for(int i = 0;i < hashTable[index].size();i++)
- {
- if(hashTable[index][i] == s)
- {
- cout << s << " is found!" << endl; return;
- } }
- cout << s << " is not found!" << endl; }

Example


Hash Table
Keys
HackerEarth
Code Monk
Tutorial
Hashing
0
Tutorial
1
2
Code Monk
3
4
HackerEarth
Hashing

Thank you