SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

COIMBATORE – 35

DEPARTMENT OF COMPUTER SIENCE AND ENGINEERING (UG & PG)

First Year, 2nd Semester

2 Marks Question and Answer

Subject Code & Name: 19ITT102 & Data Structures and Algorithms

Prepared by : Mrs.G.DEVI /AP / CSE

UNIT – V

HASHING & PRIORITY QUEUE

1. Define Hashing.

Hashing is the transformation of string of characters into a usually shorter fixed length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the short hashed key than to find it using the original value.

2. Define hashing function

A hashing function is a key-to-transformation, which acts upon a given key to compute the relative position of the key in an array.

A simple hash function

HASH(KEY_Value)= (KEY_Value)mod(Table-size)

3. What is open addressing?

Open addressing is also called closed hashing, which is an alternative to resolve the collisions with linked lists. In this hashing system, if a collision occurs, alternative cells are tired until an empty cell is found.

There are three strategies in open addressing:

- Linear probing
- Quadratic probing
- Double hashing

3. What are the collision resolution methods?

The following are the collision resolution methods

- Separate chaining
- Open addressing
- Multiple hashing

4. Define separate chaining

It is an open hashing technique. A pointer field is added to each record location, when an overflow occurs, this pointer is set to point to overflow blocks making a linked list. In this method, the table can never overflow, since the linked lists are only extended upon the arrival of new keys

6. What do you mean by hash table?

The hash table data structure is merely an array of some fixed size, containing the keys. A key is a string with an associated value. Each key is mapped into some number in the range 0 to tablesize-1 and placed in the appropriate cell.

7. What do you mean by hash function?

A hash function is a key to address transformation which acts upon a given key to compute the relative position of the key in an array. The choice of hash function should be simple and it must distribute the data evenly. A simple hash function is hash_key=key mod tablesize.

8. Write the importance of hashing.

- Maps key with the corresponding value using hash function.
- Hash tables support the efficient addition of new entries and the time spent on searching for the required data is independent of the number of items stored.

9. What do you mean by collision in hashing?

When an element is inserted, it hashes to the same value as an already inserted element, and then it produces collision.

10. What are the collision resolution methods?

- Separate chaining or External hashing
- Open addressing or Closed hashing

11. What do you mean by separate chaining?

Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value. This is called separate chaining because each hash table element is a separate chain (linked list). Each linked list contains all the elements whose keys hash to the same index.

12. Write the advantage of separate chaining.

- More number of elements can be inserted as it uses linked lists.

13. Write the disadvantages of separate chaining.

- The elements are evenly distributed. Some elements may have more elements and some may not have anything.
- It requires pointers. This leads to slow the algorithm down a bit because of the time required to allocate new cells, and also essentially requires the implementation of a second data structure.

14. What do you mean by open addressing?

Open addressing is a collision resolving strategy in which, if collision occurs alternative cells are tried until an empty cell is found. The cells h0(x), h1(x), h2(x),…. are tried in succession, where hi(x)=(Hash(x)+F(i))mod Tablesize with F(0)=0. The function F is the collision resolution strategy.

15. What are the types of collision resolution strategies in open addressing?

- Linear probing
- Quadratic probing
- Double hashing

16. What do you mean by Probing?

Probing is the process of getting next available hash table array cell.

17.What do you mean by linear probing?

Linear probing is an open addressing collision resolution strategy in which F is a linear function of i, F(i)=i. This amounts to trying sequentially in search of an empty cell. If the table is big enough, a free cell can always be found, but the time to do so can get quite large.

18. What do you mean by primary clustering?

In linear probing collision resolution strategy, even if the table is relatively empty, blocks of occupied cells start forming. This effect is known as primary clustering means that any key hashes into the cluster will require several attempts to resolve the collision and then it will add to the cluster.

19. What do you mean by quadratic probing?

Quadratic probing is an open addressing collision resolution strategy in which F(i)=i2. There is no guarantee of finding an empty cell once the table gets half full if the table size is not prime.This is because at most half of the table can be used as alternative locations to resolve collisions.

20. What do you mean by secondary clustering?

Although quadratic probing eliminates primary clustering, elements that hash to the same position will probe the same alternative cells. This is known as secondary clustering.

21.What is Binary Heap?

A Binary Heap is a complete Binary Tree which is used to store data efficiently to get the max or min element based on its structure.

22. Operations on Heap:

Below are some standard operations on min heap:

getMin(): It returns the root element of Min Heap. The time Complexity of this operation is O(1). In case of a maxheap it would be getMax().

extractMin(): Removes the minimum element from MinHeap. The time Complexity of this Operation is O(log N) as this operation needs to maintain the heap property (by calling heapify()) after removing the root.

decreaseKey(): Decreases the value of the key. The time complexity of this operation is O(log N). If the decreased key value of a node is greater than the parent of the node, then we don't need to do anything. Otherwise, we need to traverse up to fix the violated heap property.

insert(): Inserting a new key takes O(log N) time. We add a new key at the end of the tree. If the new key is greater than its parent, then we don't need to do anything. Otherwise, we need to traverse up to fix the violated heap property.

delete(): Deleting a key also takes O(log N) time. We replace the key to be deleted with the minimum infinite by calling decreaseKey(). After decreaseKey(), the minus infinite value must reach root, so we call extractMin() to remove the key.

23.Applications of Heaps

Heap Sort: Heap Sort uses Binary Heap to sort an array in O(nLogn) time.

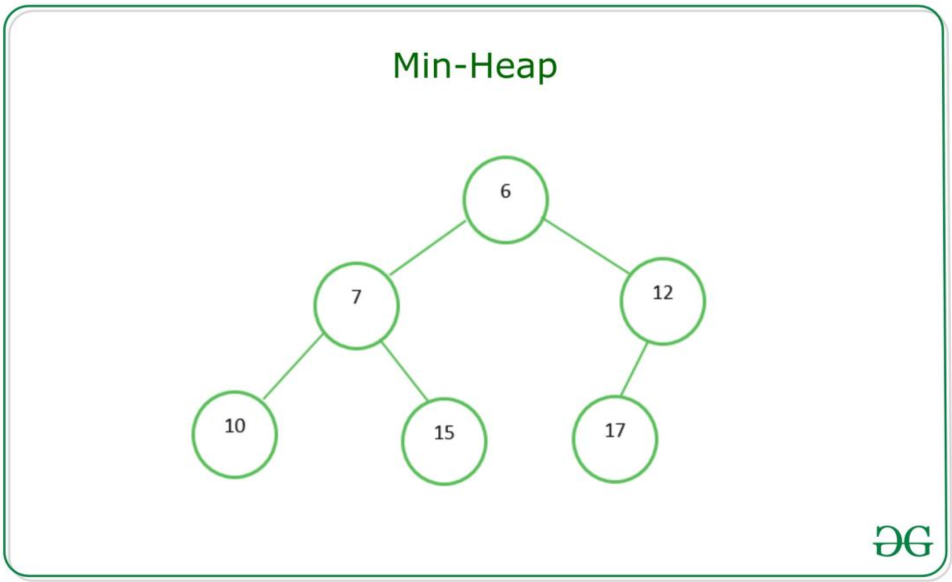
Priority Queue: Priority queues can be efficiently implemented using Binary Heap because it supports insert(), delete() and extractmax(), decreaseKey() operations in O(log N) time. Binomial Heap and Fibonacci Heap are variations of Binary Heap. These variations perform union also efficiently.

Graph Algorithms: The priority queues are especially used in Graph Algorithms like Dijkstra's Shortest Path and Prim's Minimum Spanning Tree.

24. Define Min-Heap and Max Heap

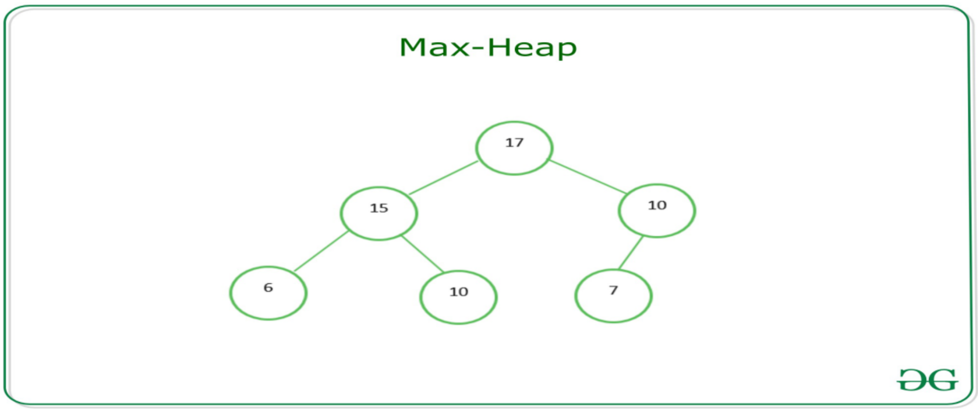
Min-Heap

In a Min-Heap the key present at the root node must be less than or equal among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree. In a Min-Heap the minimum key element present at the root. Below is the Binary Tree that satisfies all the property of Min Heap.



Max Heap

In a Max-Heap the key present at the root node must be greater than or equal among the keys present at all of its children. The same property must be recursively true for all sub-trees in that Binary Tree. In a Max-Heap the maximum key element present at the root. Below is the Binary Tree that satisfies all the property of Max Heap.



25. Difference between Min Heap and Max Heap

	Min Heap	Max Heap
1.	In a Min-Heap the key present at the root node must be less than or equal to among the keys present at all of its children.	In a Max-Heap the key present at the root node must be greater than or equal to among the keys present at all of its children.
2.	In a Min-Heap the minimum key element present at the root.	In a Max-Heap the maximum key element present at the root.

	Min Heap	Max Heap
3.	A Min-Heap uses the ascending priority.	A Max-Heap uses the descending priority.
4.	In the construction of a Min-Heap, the smallest element has priority.	In the construction of a Max-Heap, the largest element has priority.
5.	In a Min-Heap, the smallest element is the first to be popped from the heap.	In a Max-Heap, the largest element is the first to be popped from the heap.

26.Given an array representation of min Heap, convert it to max Heap.

Examples: Input: arr[] = {3, 5, 9, 6, 8, 20, 10, 12, 18, 9}

```
           3
         /    \
       5       9
     /  \   / \
   6     8  20   10
  /  \  /
12   18 9
```

Output: arr[] = {20, 18, 10, 12, 9, 9, 3, 5, 6, 8}

```
        20
      /   \
   18      10
  /   \   / \
 12     9  9    3
/  \   /
5    6 8
```

Input: arr[] = {3, 4, 8, 11, 13}

Output: arr[] = {13, 11, 8, 4, 3}