Date 22/11/23
Page No. 46

ASSIGNMENT-5

Q1) A hash table contains 10 buckets and uses linear probing to resolve collisions. The key values are integers and the hash function used is key %. 10. If the values 43, 165, 62, 123, 142 are inserted in the table, in what location would the key value 142 be inserted?

Answer) Given :-

- The hash table contains 10 buckets.
- The hash function used is key %. 10.
- The values 43, 165, 62, 123 and 142 are inserted in the table.

Calculations :-

- The hash value for the given keys are :-

43 %. 10 = 3

165 %. 10 = 5

62 %. 10 = 2

123 %. 10 = 3

142 %. 10 = 2 (Collision with 62, probe to the next slot)

142 will be inserted in the next available slot is 6.

Answer -

The key value 142 will be inserted in location 6.

Q2) Write the advantage of chained hash table (external hashing) over the open addressing scheme.

Answer) In an open addressing scheme sometimes though element is present we can't delete it if empty buckets comes in between while searching for that element, where as external hashing is free from the limitations.

Q3) Explain Graph Traversal Algorithms? With suitable examples.

Answer) The graph has 2 types of traversal algorithms. These are called the Breadth First Search and Depth First Search.

Breadth First Search (BFS) -
The BFS traversal is an algorithm, which is used to visit all of the nodes of the given graph.

Algorithm :-

BFS(Vertices, Start)
Input: The list of vertices and the start verten
Output: Traverse all the nodes, if the graph is connected.

Begin
    Define an empty queue.
    At first mark all nodes status
    as unvisited.

Add the start verten into the queue.
While queue is not empty, do delete item from queue and set to u.
Display the verten u.
For all vertices I adjacent with u, do
if vertices [i] is unvisited
Then, mark vertices [i] as temporarily visited
add v into the queue.
done
mark u as completely visited.
Done.
End.

Depth First Search (DFS) -
The Depth First Search is a graph traversal algorithm. In the algorithm one starting verten is given, and when an adjacent verten is found it moves to the adjacent verten first and try to traverse in the same manner.

Algorithm :-

① Set all the vertens to infinity, excluding the source verten.
② Push the source in the form (distance, verten) and put it in the min-priority queue.
③ From the priority queue pop out the minimum distant verten from the source verten.

(4) Update the distance after popping out the minimum distant vertex and calculate the vertex distance using (vertex distance + weight < following vertex distance.

(5) If you find that the visited vertex is popped, move ahead without using it.

(6) Apply the steps until the priority queue is found to be empty.

Q4) Give comparison between tree and graph.

Answer)

Graph	Tree
(1) A graph can be connected or disconnected, can have cycles or loops and does not necessarily have a root node.	(1) A tree is a type of graph that is connected acyclic and has a single root node.
(2) Each node can have any number of edges.	(2) If there is n nodes then there would be n-1 no. of edges.
(3) They can be directed or undirected.	(3) They are always directed.
(4) There is no unique node called root.	(4) There is a unique node called root nodes in trees.
(5) A cycle can be forms.	(5) There will not be any cycle.

⑥ For graph traversal, we use, Breadth first search and depth first search methods.	⑥ We traverse a tree using, preorder, inorder and postorder traversal methods.

Q5) What do you mean by hashing and collision? Discuss the advantage and disadvantage of hashing over other searching techniques.

Answer) Hashing refers to the process of generating a fixed size output from an input of variable size using the mathematical formulas known as hash functions. This technique determines an index or location for storage of an item in the data structure.

Eg:-

List = [11, 12, 13, 14, 15]

H(n) = [n % 10]

11%10 12%10 14%10 15%10

	0	1	2	3	4	5
Hash Table		11	12	13	14	15

The hashing process generates a small number for a big key, so there is a possibility that 2 keys could produce the same value. The situation where the newly inserted key maps to an already occupied, and it must be handled using some collision handling technology.

Advantages of Hashing :-

① Key value support - Hashing is ideal for implementing key value data structures.

② Efficiency - Insertion, deletion and searching operations are highly efficient.

③ Memory usage reduction - Hashing requires less memory as it allocated a fixed space for storing elements.

Q6) What are the various ways to represent graph? Find the following two for the graph given below in Q7:
(i) Adjacency list representation.
(ii) Adjacency matrix representation.

Answer) A graph is a non linear data structure consisting of vertices and edges, the vertices are sometimes also refred to as nodes and the edges are lines or arcs that connect any 2 nodes in a graph.

Representation of a graph - The 2 most common ways of representing a graph:
(1) Adjacency matrix
(2) Adjacency list

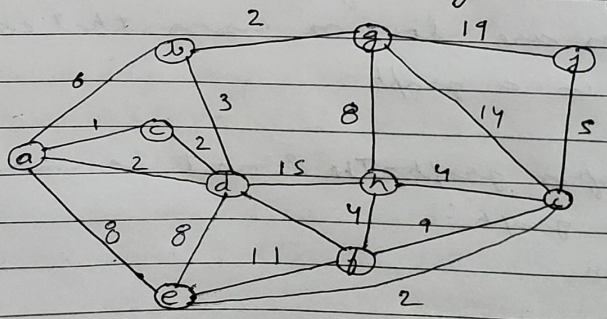
(1) Adjacency matrix - It is a way of representing a graph as a matrix of boolean (0's and 1's). Let's assume there are n vertices in the graph. So, create a 2D matrix adjMat[n][n] having dimension n x n.

★ If there is an edge from vertex i to j, mark adjMat[i][j] as 1.

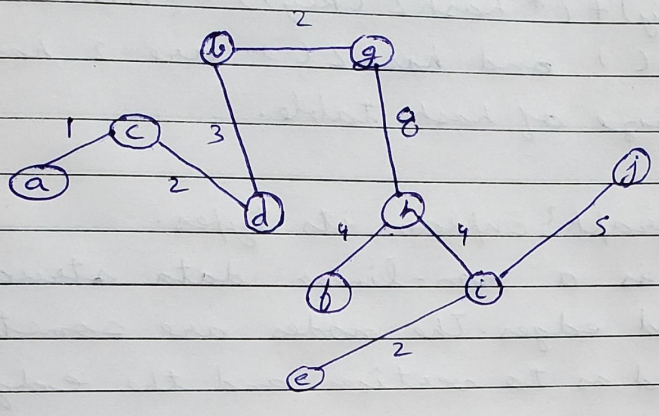
★ If there is no edge from vertex i to j, mark adjMat[i][j] as 0.

(2) Adjacency List - An array of lists is used to store edges between two vertices. The size of array is equal to the no. of vertices (n, n). Each index in this array represents a specific vertex in the graph. The entry at the index i of the array contains a linked list containing the vertices that are adjacent to vertex i.

(07) Using prim's and kruskal's algorithm, find the MST from the following graph. What is the weight of a minimum spanning tree of the following graph?



wer) After applying Prim's algorithm, start from A is shown in figure below.



Adding all the weights in the given figure which will be equal to 31.

$$1+3+2+2+4+8+4+2+5 = 31$$

Q8) Explain Double Hashing.

Answer) Double hashing is a collision resolution technique used in hash tables:

★ It works by using two hash functions to compute two different hash values for a given key.

★ The first hash function is used to compute the initial hash value, and the second hash function is used to compute the step size for the probing sequence.

☆ Double hashing can be done using:

(hash1(key) + i * hash2(key)) % TABLE-SIZE.
Here hash1() and hash2() are 2 hash functions and Table-Size of hash table.

Q9) What is graph? Explain its types.

Answer) A graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred as to vertices and the edges are lines or arcs that connect any two nodes in graph.

Types of Graph:-

(1) Undirected Graph - A graph in which edges have no direction, i.e., the edges do not have arrows indicating the direction of traversal.

(2) Directed Graphs - A graph in which edges have a direction, i.e., the edges have arrows indicating the direction of traversal.

(3) Weighted Graphs - A graph in which edges have weights or costs associated with them.

(4) Unweighted Graphs - A graph in which edges have no weights or costs associated with them.

(5) Complete Graphs - A graph in which each vertex is connected to every other vertex. Eg - A tournament graph where every player plays against every other player.