Assignment-5

Name - Vibha Rani
Roll no - 45
Section - CE [Computer Engineering]
University Roll no - 2017395

Tutorial - 05

Ques1: what is difference between DFS and BFS. Please write the application of both Algos.

→ Using BFS we can find the minimum no.s of nodes between source node and destination node, while using DFS, we can find if a path exists between two nodes

→ Applications of DFS:-

1. Detecting cycles in graph
2. Finding path between two given vertices u and v.
3. If we perform, DFS on unweighted graph, then it will create minimum spanning tree for all pair shortest path tree.
4. Topological sorting can be done using DFS.

→ Applications of BFS:

1. Like DFS, BFS may also be used for detecting cycles in graph.
2. Finding shortest path and minimum spanning tree in unweighted graph.
3. Finding route through GPS navigator system with minimum no. of crossing.
4. In Networking, finding route for packet transmission.

Ques2: Which data structure are used to implement BFS & DFS & why?

- DFS uses stack data structures as order doesn't has much importance.
- BFS uses queue data structure as order matters in this case.

Ques3: What do you mean by sparse & dense graph? Which representation of graph is better for sparse & dense graphs?

→ Sparse graph – Graph in which no. of edges is much less than the possible no. of edges.

Dense graph – Graph in which no. of edges is close to the maximal no. of edges.

If the graph is sparse, we should store it as a list of edges. Alternatively, if it is dense, we should store it as adjacency matrix.

Ques4: How can you detect a cycle in a graph using DFS and BFS?

Ans Using BFS:

(i) Compute in degree no of incoming edges for each of the vertex present in graph and count no. of nodes = 0

(ii) Pick all vertices with in degree as 0 and add them to queue.

(iii) Remove a vertex from the queue. Then increment count by 1, and decrease in degree by 1 for all neighbours. If in-degree of a neighbouring node is 0, add to queue.

(iv) Repeat step 3 until queue empty.

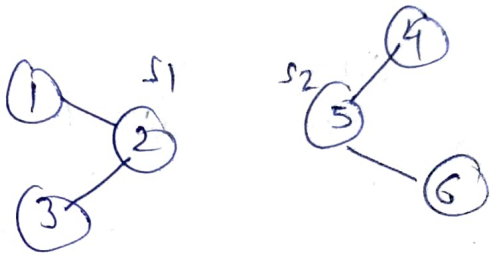
(v) If no of visited nodes is not equal to no. of nodes then graph has a cycle.

Using DFS:- Similar process is done is DFS as well. But in DFS, we have the option of doing recursive call for vertices which are adjacent to the current node and are not get visited. if recursive function returns false, then graph does not have a cycle.

Ques 5: what do you mean by disjoint set data structure? Explain 3 operations along with examples, which can be performed on disjoined sets.

→ It allows to find out whether the 2 elements are in the same set or not efficiently. The disjoint set can be divided as the subsets where there is no common element between 2 sets.

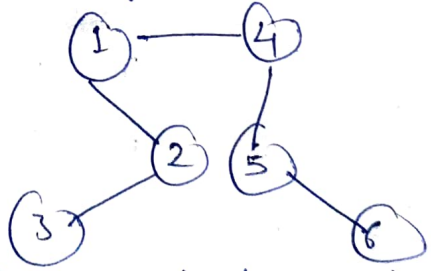
eg - $S_1 = \{1,2,3\}$
$S_2 = \{4,5,6\}$



operations:-

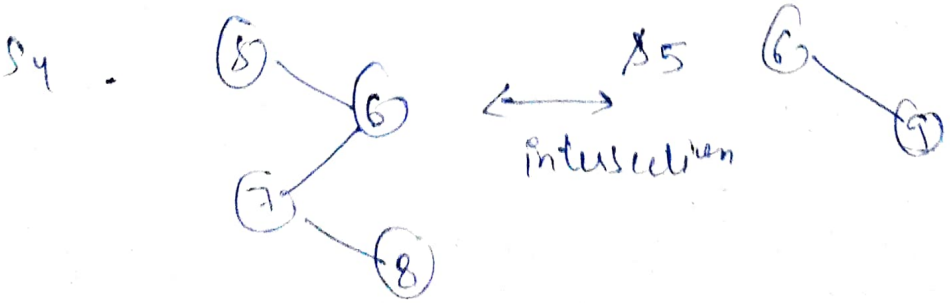
1) Union:- Merge 2 sets when edge is added

$S_1 \cup S_2 \Rightarrow S_3 \Rightarrow$

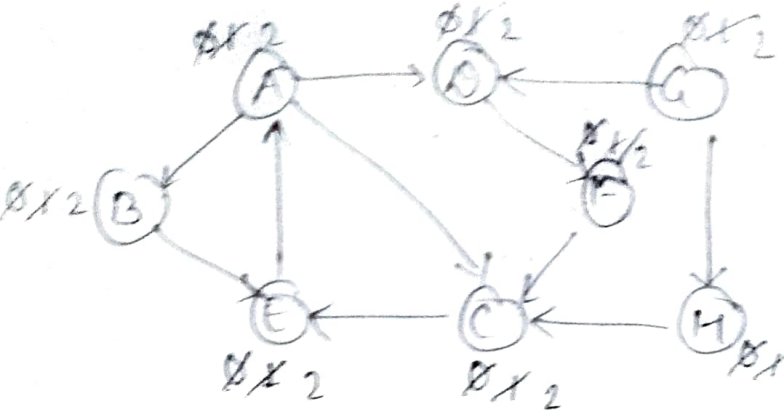


2) Find ( ):- tells which element belongs to which set
find(1) = S_1 find(5) = S_2

3) Intersections - Outputs another set as common elements
$S_1 \cap S_2 = \{\phi\}$ find $S_4 \cap S_5 = \{6\}$



Ques6: Run BFS and DFS on given graph

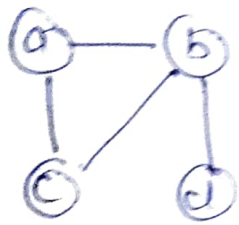


BFS:- nodes G H F D C E
A B
Parent x G G G H C E A.
Visited nodes → G H F D C E
A B
path → G → H → C → E → A → B

DFS:
nodes processed :- G G D C E A B
Stack G DFH CFH EFH AFH BFH FH
path → G → D → C → E → A → B

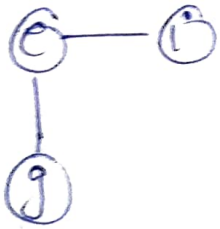
Ques7: find out no. of connected components & vertices in each comp. using disjoint set data structure

1)

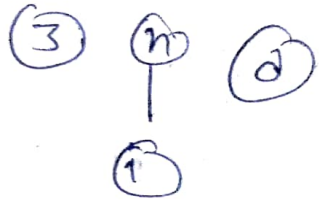


no of(V) = 4
no. of(CC) = 1

2)

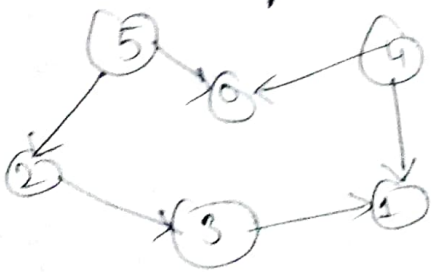


no.(V) = 3
no.(CC) = 1



no of(V) = 3
no. of(CC) = 2

Ques8: Apply topological sorting & DFS on graph starting vertex 0 to 5.



Topological: 5, 4, 2, 3, 1, 0
Adjacency List:
0 →
1 →
2 → 3
3 → 1
4 → 0, 1
5 → 2, 0

Stack | 0 | 1 | 3 | 2 | 4 | 5 |
DFS stack → | 4 | 0 | 1 | 3 | 2 | 5 |

Head → DFS → 5 → 2 → 3 → 1 → 0 → 4

Ques 9 Heap data structure can be used to implement priority queue? Name few graph algos where you need to use priority queue & why?

→ We can use heaps to implement priority queue. It will take O(logn) time. to insert & delete each element in the priority queue. Based on heap structure, priority queue has also 2 types - max & min. Priority queue: some algos where we need to use priority queue:

1) Dijkstra's shortest path algo using priority queue - When graph is stored in the form of adjacency list or matrix, priority queue can be used extract minimum efficiently when implementing Dijkstra's algorithm.

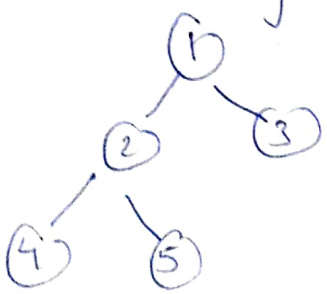
2) Prim's algorithm:- Priority queue is used to implement prim's to store key's of nodes & extract minimum key node at every step.

3) Data compression:- Priority queue is used in Huffman's code which is used to compress data.

Ques 10 what is difference between max & min heap?

→ In Min heap, the key present at the root node must be smaller than among the keys present at all of its children.

eg



In Max heap, the key present at the root node must be greater than among the key present at all of its children