Name - Vansh Kalra
Section - CST
Class Roll No. - 25
Univ Roll No. - 2017576

DATE
PAGE

Tutorial 5

Q1. What is the difference b/w DFS & BFS. Please write applications of both the algorithms.

A1.

BFS	DFS
i). Stands for Breadth first Search.	i). Stands for Depth first Search.
ii). BFS uses queue data structure.	ii). DFS uses stack data structure.
iii) BFS can be used to find single source shortest path in an unweighted graph because in BFS, we reach a vertex with min. no. of edges from a source.	iii). In DFS, we might traverse through more edges to reach a destination vertex from a source.
iv). Siblings are visited before the children.	iv). Children are visited before the siblings.
v). In BFS, there is no concept of backtracking.	v). DFS algorithm is a recursive algorithm that uses the idea of backtracking.

Applications of BFS - It is used in bipartite graph & shortest path.

Applications of DFS - It is used in acyclic graph & topological order.

Q2. Which Data Structures are used to implement BFS & DFS & why?

A2 Queue data structure is used in BFS because it works on the strategy of visiting all neighbouring nodes at present level before moving to the next level.

Stack data structure is used in DFS because it explores the nodes as far as possible (depth-wise) before being forced to backtrack & explore other nodes.

DATE _____

PAGE _____

Q3. What do you mean by sparse & dense graphs? Which representation of graph is better for sparse & dense graphs?

Ans. When there are a large number of edges in a graph then it is called a dense graph while when there are less number of edges in a graph then it is called sparse graph.

Adjacency list is better for sparse graphs.
Adjacency matrix is better for dense graphs.

Q4. How can you detect a cycle in graph using BFS & DFS?

Ans. i) We will run BFS/DFS on a graph & make the visited of that node TRUE & initialise its parent with the node from which it is called.

ii). ~~We~~ If we encounter any node which is already visited & now it is visited from any other node than its parent then we can say that cycle is present in the graph otherwise not.

Q5. What do you mean by disjoint set data structure? Explain 3 operations along with examples, which can be performed on disjoint sets.

Ans. The disjoint set data structure is also known as union-find data structure & merge-find set. It also allows to find out whether two elements are in the same set or not efficiently. The disjoint set can be defined as the subsets where there is no common element b/w the two sets.

Disjoint-set data structures support 3 operations:

i). Making a new set containing a new element.
ii). Find representative of the set containing a given element.
iii). Merging two sets.

DATE _____

PAGE _____

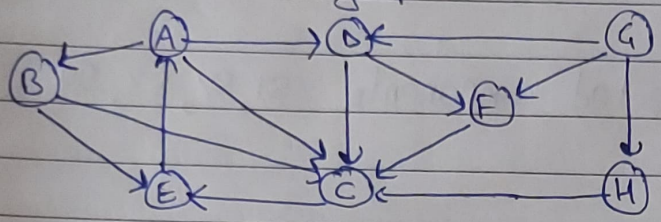
Eg -

S1 - {1}, S2 - {2}, S3 = {3}, S4 = {4} & S5 = {5}

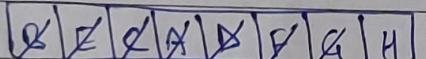
Union (S3, S4) → S3 = {3, 4}

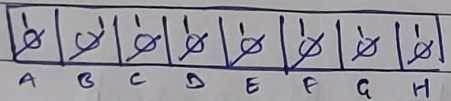
Find(4) will return representative of set S3 i.e. Find(4) = 3.

Q6 Run BFS & DFS on graph.



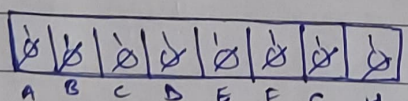
BFS

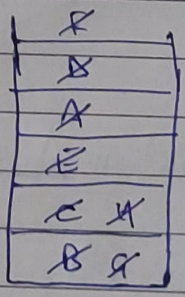
q 

visited 

B E C A D F G H

DFS

visited 

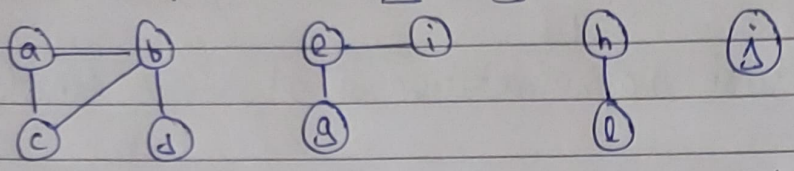


Stack

B C E A D F G H

DATE
PAGE

Q7. Find out the number of connected components & vertices in each component using disjoint set data structure.



S1 S2 S3 S4

Ans. There are 4 connected components : S1, S2, S3, S4.

No. of vertices

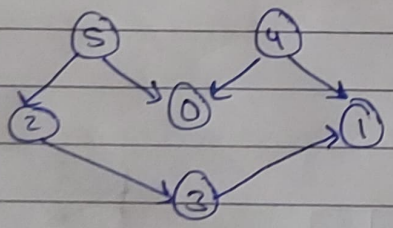
S1 = 4 {a, b, c, d}.

S2 = 3 {e, g, i}.

S3 = 2 {h, l}.

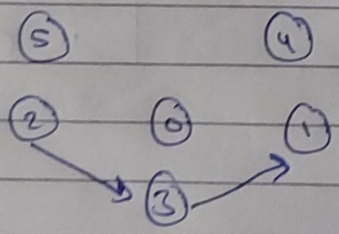
S4 = 1 {j}.

Q8. Apply topological sorting & DFS on graph.

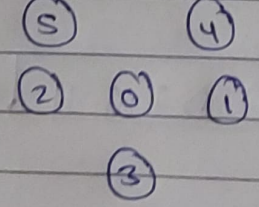


Topological sorting

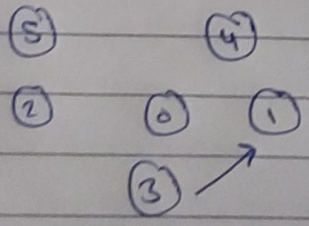
i)



iii)



ii)



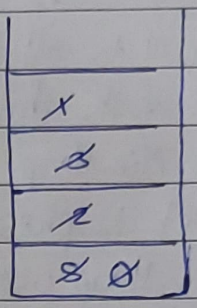
DATE _____

PAGE _____

5, 4, 2, 0, 3, 1

DFS

5, 2, 3, 1, 0, 4



Q9. Heap data structure can be used to implement priority queue? Name few graph algorithms where you need to use priority queue & why?

A9. Priority queue is similar to queue where we insert an element from the back & remove an element from front, but the logical order of elements in priority queue depends on the priority of elements. We can use heaps to implement the priority queue. It will take O(log n) time to insert & delete each element in the priority queue.

Graph algorithms where we need priority queue are:

i). Djikstra's Algorithm - To find the shortest path b/w nodes in a graph.

ii). Prim's Algorithm - To find the Minimum Spanning Tree in a weighted undirected graph.

DATE _____

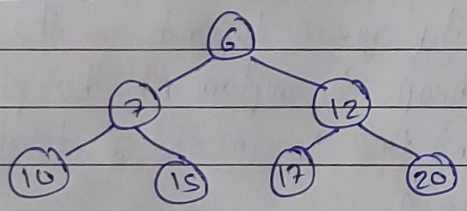
PAGE _____

Q10. What is the difference b/w Max & Min heap?

Ans10.

Min Heap	Max heap
i). In a min-heap the key present at the root node must be less than or equal to all the keys present at all of its children.	i). In a Max-heap, the key present at the root node must be greater than or equal to among all keys present at all of its children.
ii) In a Min-Heap, the minimum key element is present at root.	ii) In a Max-Heap, the maximum key element is present at root.
iii). A Min-Heap uses the ascending priority.	iii). A Max-Heap uses the descending priority.
iv) In a Min-Heap, the smallest element is the first to be popped from the Heap.	iv). In a Max-Heap, the largest element is the first to be popped from the Heap.

Min-Heap



Max-Heap