DAA

Tutorial - 5

Ayush Dhiman
Section - H
Roll No. - 28

PRADEEP
Date / /
Page No.

Ans:

	BFS	DFS
i)	Uses queue data structure.	Uses stack data structure.
ii)	Stands for breath first search.	Stands for depth first search.
iii)	Can be used to find single source shortest path in an unweighted graph, we reach a vertex with min. no. of edges from a source vertex.	We might traverse through more edges to reach a destination vertex from a source.
iv)	Siblings are visited before the children.	Children are visited before the siblings.
	Applications:	Applications:
1.	Shortest path & minimum spanning tree for unweighted graph.	Detecting cycle in a graph.
2.	Peer to peer networks.	Path finding.
3.	Social Networking websites.	Topological sorting.
4.	GPS Navigation system.	Solving puzzles with only one solution.

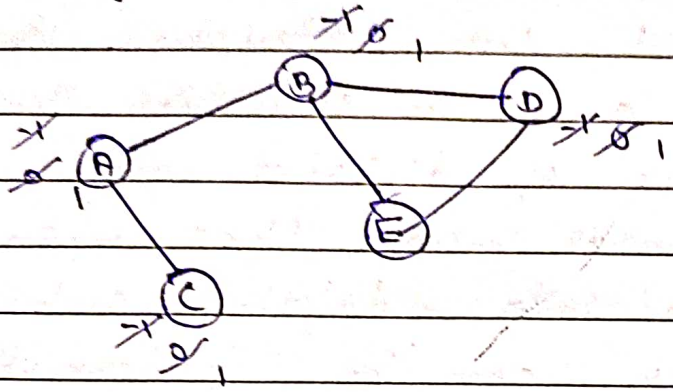
PRADEEP
Date / /
Page No.

Ans 2 In BFS we use queue data structure
queue is used when things don't ha
to be processed immediately, but ha
to be processed in FIFO order like
BFS
In DFS stack is used as DFS uses
backtracking. For DFS, we retrieve it
from root to the farthest node as
much as possible, this is the same
idea as LIFO (used by stack).

Ans 3 Dense graph is a graph in which the no
of edges is close to the maximal no.
edges
Sparse graph is a graph in which the
no. of edges is close to the minimal
no. of edges. It can be disconnected g

Adjacency lists are preferred for spar
graph &
adjacency matrix for dense graph.

Ans4 Cycle detection in undirected graph (BFS)



-1 = unvisited
0 = into the queue
1 = visited & popped out from stack

Stack :	E
	D
	B
	A

visited set :
A B C D E

=) B → D → E → B

vertex	Parent
A	-
B	A
C	B
D	C
E	D

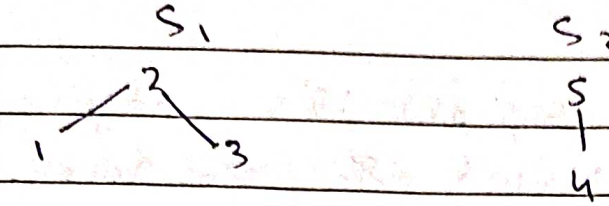
Here E finds B (adjacent vertex of E) with 0.

=) it contains a cycle.

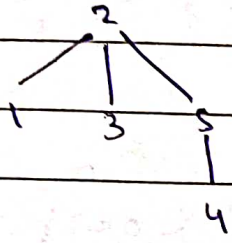
Ans 5 The disjoint set data structure is also kn as union-find data structure & merge-f set. It is a data structure that contain a collection of disjoint sets.
The disjoint set means that when the se is partitioned into disjoint subsets, various option can be performed on it. In this case, we can add new sets, we can merge the sets, & we can also find the representation member of a set It also allows to find out whether the two elements are in the same set or not efficiently.
Operation on disjoint set.
Union
a) If S_1 & S_2 are two disjoint sets, th union $S_1 \cup S_2$ is a set of all elements x such that X is in either S_1 or S_2.
b) As the sets should be disjoint $S_1 \cup S_2$ replaces S_1 & S_2 which no longer exists
c) Union is achieved by simply making one of the trees as a subtree of other i.e to set parent field of one of the roots of the trees to other root.

PRADEEP
Date / /
Page No.

eg



S1 ∪ S2

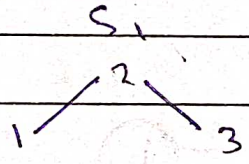


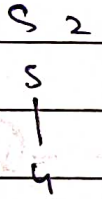
Merge the sets containing X & containing Y into one.

Find

Given an element X, to find the set containing it.

eg





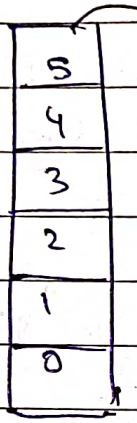
find(3) => S1

find(5) => S2

return in which set X belongs.

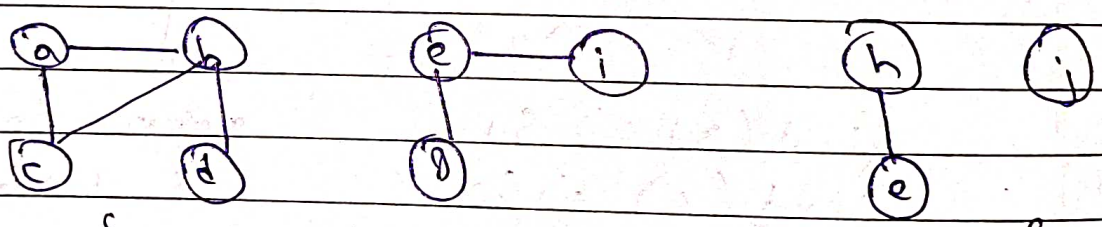
Make-Set(x): Create a set containing x

Ans 6 Go to node 5, all its adjacent nodes are already visited so push node 5 in the stack & mark it visited.



5 4 3 2 1 0
(output)

Ans 7



$V = \{a, b, c, d, e, g, h, i, j, l\}$

$E = \{(a,b), (a,c), (b,c), (b,d), (e,i), (e,g), (h,l), (j)\}$

{a} {b} {c} {d} {e} {g} {h} {i} {j} {l}

(a,b) {a,b} {c} {d} {e} {g} {h} {i} {j} {l}

(a,c) {a,b,c} {d} {e} {g} {h} {i} {j} {l}

(b,c) {a,b,c} {d} {e} {g} {h} {i} {j} {l}

(b,d) {a,b,c,d} {e} {g} {h} {i} {j} {l}

(e,i) {a,b,c,d} {e,i} {g} {h} {j} {l}

(e,g) {a,b,c,d} , {e,i,g} {h,l} {j}

(h,l) {a,b,c,d} {e,i,g} {h,l} {j}

(j) {a,b,c,d} {e,i,g} {h,l} {j}

We have

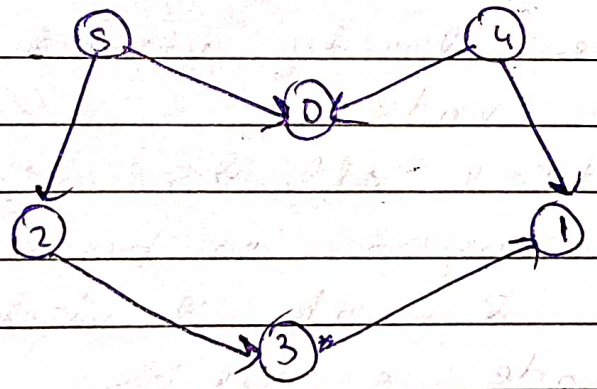
{a,b,c,d}

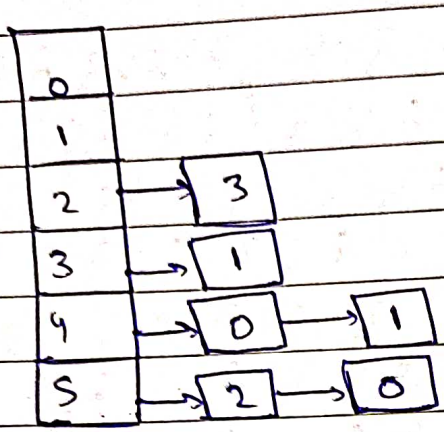
{e,i,g}

{h,l}

{j}

Ans 8

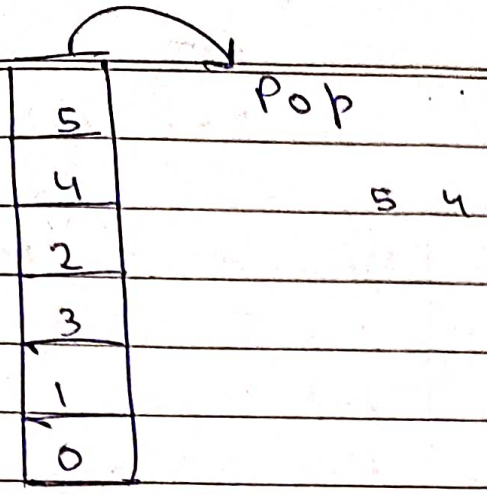




Algo:

1) Go to node 0, it has no outgoing edges so push node 0 into the stack & mark visited.
2) Go to node 1, again it has no outgoing edges so push node 1 into the stack & mark it visited.
3) Go to node 2, process all the adjacent nodes and mark node 2 visited.
4) Node 3 is already visited so continue with next node.
5) Go to node 4, all its adjacent nodes are already visited so push node 4 into the stack & mark it visited.
6) Go to node 5, all its adjacent nodes are already visited so push node 5 into stack and mark it visited.

PRADEEP
Date / /
Page No.



5 4 2 3 1 0
(output)

Ans⇒ Heap is generally preferred for priority queue implementation because heap provides better performance compared to array or linked list.

Algorithms where priority queue is used -

1) Dijkstra's shortest path Algorithm: when the graph is stored in the form of adjacency list or matrix, priority queue can be used to extract minimum efficiently when implementing dijkstra's algorithm.
2) Prim's algorithm: To store keys of nodes & extract minimum key node at every step.

PRADEEP
Date / /
Page No.

	Min heap	Max heap
1)	Your every pair of the parent & descendant child node, the parent node always has lower value than descended child node.	For every pair of the parent & descendant child node, the parent node has greater value than descended child node.
2)	The value of nodes increases as we traverse from root to leaf node.	The value of nodes decreases as we traverse from root to leaf node.
3)	root node has the lowest value.	The root node has the greatest value.