Date.
Page No.

DAA Tutorial-5

Siddhi Shankar
20021517

Q Difference b/w DFS & BFS. Please write the algorithms of both the algorithm

Ans

	DFS	BFS
1)	It stands for depth First search	It stands for Breadth first search
2)	DFS uses stack to find the shortest path	BFS uses Queue to find the shortest path.
3	It finds the shortest path to the destination	It finds all the shortest path to the destination
4.	It is implimented using LIFO list.	It is implimented using FIFO
5.	Backtracking is used in DFS	~~It is implemented using FIFO~~ No Backtracking is there

Application

1) BFS:-

- It is used for detecting cycle in a graph.
- Finding a root to GPS navigation system with minimum no. of crossing
- In networking, finding a route for packet transmission.

Date.
Page No.

Siddhi Shankar
20021517

2) DFS:

- For detecting cycles in a graph.
- It is used in the generation of topological sorting
- We can find strongly connected component of graph.

Q2) Which data structure are used to implemented BFS and DFS and why?

Ans BFS (Breadth First Search) uses queue data structure for finding the shortest path.

DFS (Depth First Search) uses stack data structure.

A queue (FIFO - First in First Out) data structure is used by BFS. You mark any node in the graph as root and start traversing the data from it. BFS traverses all the nodes in the graph and keeps dropping them as completed. BFS visits an adjacent unvisited node, marks it as done, and inserts it into a queue.

DFS algorithm traverses a graph in a depthward motion and uses a stack to remember to get the next vertex to start a search, when a dead end occurs in any iteration.

Date.
Page No.
Siddhi Shankar
20021517

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

A3 Sparse graph is defined as a graph in which the number of edges is much less than the possible number of edges.
Whereas, dense graph is a graph in which the number of edges is closed to the maximal no. of edges.

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

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

A3 For BFS

1) No of incoming edges for each of vertex present in graph and initalize the count of visited nodes as 0.

2) Pick all the vertices within degree as 0 and add them into a Queue.

3 Remove a vertex from queue.

4) Repeat step 3 until queue is empty.

5) If the count of visited nodes is not equal to the no. of nodes in the graph has cycle otherwise not.

Date.
Page No.
Siddhi Shankar
20021517

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

A3 Disjoint set in defined as the subset where there is no common element b/w the two sets.

$S1 = \{1, 2, 3, 4\}$
$S2 = \{5, 6, 7, 8\}$

let $S1 \cup S2 = \{1, 2, 3, 4, 5, 6, 7, 8\}$
$S_3 = \{1, 2, 3, 4, 5, 6, 7, 8\}$

Operations:-

1) Making new sets: The (make set) operation adds a new elements into newest containing only the new element, & new

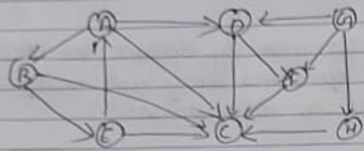
2) Merging two sets: The operation (union(x,y)) replace the set containing x & set y with their union

3) Finding set Representation: The (find) operation follows the chain of parent pointers from a specific query of node x until it reaches a root element. This root element represents the set to which x belongs & may be

Date.
Page No.
Siddhi Shankar
2020451?

x itself. (and) returns the next elements it reaches

Q Run BFS & DFS on graph shown on right side



Ans) let source node → A & goal node → F.

* For BFS → A

{B, C, D}	{A}
{D, C, E}	{A, B}
{C, E, F}	{A, B, D}
{E, F}	{A, B, D, C}
{F}	{A, B, D, C, E}

→ {A B D, C, E, F}

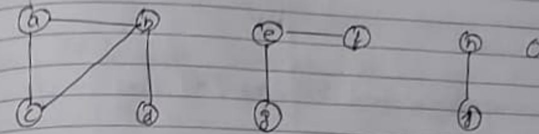
For DFS

Visited	A	B	D	C	E	F
Stack	B	E	F	E	F	
	D	D	E	F		
	C	C				

⇒ {A, B, D, C, E, F}

Date.
Page No.
Siddhi Shankar
20204517

Q Find out the number of connected components and vertices in each components using disjoint set data structure.

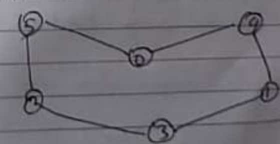


Ans)

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

Number of connected components = 3 ans.

Q) Any topological sorting & DFS on graph having vertices from 0 to 5?



Date.
Page No. Siddhi Shankar 2002451

43) 0 →
1 →
2 → 3
3 → 1
4 → 0, 1
5 → 2 0

	0	1	2	3	4	5
visited	false	false	false	false	false	false

Stack (empty)

① Topological sort (0) visit [0] = true

stack [0]

② Topological sort (1) visited [1] = true

stack [0 | 1]

③ Topological sort (2) visited [2] = true

Topological sort (3) visited [3] = true
stack [0 | 1 | 2 | 3]

④ Topological sort (4) visited [4] = true
as [0 & 1] already visited

stack [0 | 1 | 2 | 3 | 4]

⑤ Topological sort (5) visited [5] = true
as [2, 0] already visited

stack [0 | 1 | 2 | 3 | 4 | 5]

→ [0 | 1 | 2 | 3 | 4 | 5]

Date.
Page No. Siddhi Shankar 2002451

DFS can be '543210' but it is not a topological sort.

In DFS, we print a vertex & then recursively call DFS for its adjacent vertexes.

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

→ Heap data structure can be used to implement priority queue. As a priority queue is different from normal queue because instead of FIFO, values are come out in order of priority

Some algo's are

i) Dijkstra's algorithm
ii) Prim's

6) Difference between max and min heap

min heap	max heap
i) In this key present at root node must be less than or equal to keys present	In this key present at root node must be greater than or equal to key present
In this minimum keys are present at root node	In this maximum key are present at root node.

Date.
Page No.

It uses according priority	It uses the descending priority
In this smallest element has priority	In this largest element has priority
In this smallest element is first to be popped from heap	In the largest element is first to be popped from heap