

GRAPHSAIM

Write a menu driven C program to perform the following operations on a directed graph.

- (i) in degree of a particular node.
- ii) out degree of a particular node
- iii) DFS
- iv) BFS

v) Display (using Adjacency List and Adjacency matrix)

ALGORITHM

- 1 Start
- 2 Enter no. of vertices
- 3 Enter the vertices
 - 3.1 Add the vertices to vertices array and vertices-list
- 4 Repeat until $i < n$ & $j < n$
 - 4.1 check if an edge is present from i th vertex to j th vertex
 - 4.1.1 set $adj_matrix[i][j] = 1$
 - 4.1.2 Add vertex to end of vertices-list $[i]$
 - 4.2 Else set $adj_matrix[i][j] = 0$
- 5 Enter choice for menu
- 6 case 1: in degree of a node
 - 6.1 Enter vertex, set $count = 0$
 - 6.2 Search vertex in vertices array, set $count++$ if

6.3 Repeat until $j < n$

6.3.1 set $count = count + adj_matrix[i][j]$

7. Case 2: Find ~~out~~ degree of a node

7.1 Enter vertices, set $count = 0$

7.2 search vertices in vertices-array, set location

7.2.1 if not found, return

7.3 Repeat until $j < n$

7.3.1 set $count = count + adj_matrix[i][j]$

8. Case 3: BFS

8.1 Enter starting vertex, set $flag = 0$

8.2 set $status = 1$ for all elements in vertices-list

8.3 search for vertex in vertices-list, set $flag = 1$, $i = location$

8.3 no not found, return

8.4 Insert into queue vertices-list- $[i]$

8.5 set $status = 2$ for vertices-list- $[i]$

8.6 Repeat until $front \neq -1$

8.6.1 set $ptr = element$ after deletion from front

8.6.2 print vertex

8.6.3 set $status = 3$ for printed vertex

8.6.4 Repeat until $ptr \rightarrow next = NULL$

8.6.4.1 insert element into queue

8.6.4.2 set inserted element $status = 1$

9. Case 4: DFS

9.1 Enter starting vertex, set $flag = 1$

9.2 set $status = 1$ for all elements in vertices-list

9.3 Search for vertex in vertices-list, $beg = 1$, $i = \text{loc_curr}$

9.3.1 If not found, return

9.4 push vertex onto stack, set status = 2.

9.5 Repeat until top = 1

9.5.1 set $ptr = \text{element above top}$

9.5.2 Print vertex, set status = 3

9.5.3 Repeat until $ptr \rightarrow \text{next} = \text{NULL}$

9.5.3.1 push element into stack with status = 1

9.5.3.2 set pushed element status = 2

10 Case 5: Display adjacent matrices

10.1 Repeat until $i \leq n$

10.1.1 Repeat until $j \leq n$

10.1.1.1 Print $\text{adj-matrix}[i][j]$

11.5 Case 6: Display adjacency list

11.1 Repeat until $i \leq n$

11.1.1 set $ptr = \text{vertices-list}[i]$

11.1.1.1 print vertex

11.1.1.2 print list connected to vertex

12 Stop

CONCLUSION

The program has been executed correctly and output has been verified