

Tutorial & Laboratory

Programming & Data Structure: CS11001/19001

Section - 4/D

DO NOT POWER ON THE MACHINE

Department of Computer Science and Engineering I.I.T.
Kharagpur

Spring Semester: 2013 - 2014 (17.04.2014)

Download

**Download the file date170414.pdf from
Programming & Data Structures ... of**

<http://cse.iitkgp.ac.in/~goutam>

**View the file using the command `acroread` & or
`xpdf` &**

Assignment XVII

Write a C program to perform the following tasks. [Marks: 5 + 5 + 5 + 5]

Task I

The function `int main(int ac, char *av[])` takes two arguments. Pass an input file name as command line argument. The name of the input file is available in `av[1]`. Open it in `read mode`. The input file contains data of a `undirected graph`.

Task I (cont.)

Read the number of vertices v from the opened file and create an array `char adj[v][v]`. Read the edges from the file and build the **adjacency matrix** in `adj[v][v]`. If there is an edge between the nodes $\{i, j\}$, then both `adj[i][j]` and `adj[j][i]` should be '1', otherwise these entries are '0's. [Marks: 5]

Sample Input

5

0 1 0 4 0 2 1 2 1 4 2 3 2 4 3 4

The graph is $G = (V, E) =$
 $(\{0, 1, 2, 3, 4\}, \{\{0, 1\}, \{0, 4\}, \{0, 2\}, \{1, 2\}, \{1, 4\},$
 $\{2, 3\}, \{2, 4\}, \{3, 4\}\})$.

Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	1
1	1	0	1	0	1
2	1	1	0	1	1
3	0	0	1	0	1
4	1	1	1	1	0

Task II

Read in `main()` two vertices s (source) and d (destination) from the keyboard. Call the function `int path(int v, char adj[][v], int s, int d)` that will return the length of the path, if there is one from $s \rightsquigarrow d$ in the graph, otherwise it will return `return 0`.

Task II (cont.)

The function `path()` calls `void dfs(int v, char adj[][v], char visited[], int parent[], int s)` for *depth-first traversal* of the graph starting from `s`.

The function `path()` also prints the path by calling the function `int printPath(int parent[], int dst)`, where `parent[]` information of each node traversed is filled in `dfs()`. [Marks: 5]

Sample Input

5

0 1 0 4 0 2 2 4

Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	1
1	1	0	0	0	0
2	1	0	0	0	1
3	0	0	0	0	0
4	1	0	1	0	0

Path

Enter source and destination: 1 4

Path: 1 0 2 4

Length: 3

Enter source and destination: 1 3

No path from 1 to 3

Task III

Calls the function `int component(int v, char adj[][v])` from `main()` which returns the number of components of the graph. [Marks: 5]

Sample Input

5

0 1 0 4 0 2 2 4

Components

Components: 2

Task IV

Calls the function `int cycles(int v, char adj[][v])` that returns the number of fundamental cycles present in the graph. It also prints each of these fundamental cycles.

[Marks: 5]

Sample Input

5

0 1 0 4 0 2 1 2 1 4 2 3 2 4 3 4

Fundamental Cycles

Cycle: 2 1 0

Cycle: 4 3 2 1 0

Cycle: 4 3 2 1

Cycle: 4 3 2

No. of Cycles: 4

```
$ ftp 10.5.17.186
Connected to 10.5.17.186.
220----- Welcome to Pure-FTPd ----
220-You are user number 1 of 50 allowed.
220-Local time is now 07:54. .... 21.
220-IPv6 connections .....
220 ... disconnected .. inactivity.
Name (10.5.17.186:..): pds
```

Submission by ftp

```
331 User pds OK. Password required
Password: pds04
230-User pds has group access to: pds
230 OK. Current restricted directory is /
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd assignment17
250 OK. Current directory is /assignment17
```

Submission by ftp

```
ftp> put D0617.c
local: D0617.c remote: D0617.c
200 PORT command successful
150 Connecting to port 47093
226-File successfully transferred
226 0.001 seconds .. 39.00 Kbytes ..
27 bytes sent in 0.00 secs (1098.6 kB/s)
ftp> bye
21-Goodbye. ....
221 Logout.
$
```