

Date: / / 20

had soutanted





341.70	Name - Shontany Deshpande
	class - SycsE
	Rollno7
	Batch - St
N 04/20 V	Sub-7/0. Sub-100 sett add observed
	ot or lossoits is to wantege diside so ted of and
-37,04	Practical No. 7 De Louis
	transite at a sample on a town a marriage
	Aim: Implementation OF DES and BES
9,6	note and and to the tot the color of
	Theory:
	(waph:
00000	A graph Gran be defined as a pair (VE), where
0.0 .22	Vis a set of vertices, and E is a set of edges between
	the vertices E.
	course and larger of and I style taxonala act
2,014	Directed graph: A graph whose edges are ordered pairs
6 60	adamente tood of vertices some and on
	dans batanch a aprime of
	Undirected graph: A graph whose edges are unordered
	pairs of vertices dans
01 060	MISTER CONTRACT CONTRACT
pibar	(4) (5) (D)
mont	and truly dealined at a little of above and country
9000	warmy too ab sec 3) as 2 garden of days
	mount show

Page No. 1



A group of academic Research Institutions

Date: / /20



There are different ways to store graphs in a computer

1. Adjacency list +

Much like the incidence list, each verten has a list of which vertices it is adjacent to. This caused redundancy in an undirected graph: for example, if vertices A and B are adjacent, A's adjacency list contain B, while B's list contains A. Adjacency queries are faster, at the cost of entra storage space.

2. Adjacency matrin +

This is the n by n matrin A, wheren is the number of vertices in the graph. If there is an edge from some verten n to some verten y, then the element any is I (or in general the number of ny edges) atherwise it is o. In computing, this matrin makes it easy to find subgraphs, and to reverse a directed graph.

· Graph Traversal Techniques ?

graph enactly once. Because there are many paths leading from one mode to another, the hardest part about traversm a graph; making sure that you do not process some node twice.



Date:

/20

Date . / / ZU Agroup of academic Research Institutions quality For Excellent



2000	1. Depth - first search : 100 and though (1)
	boding of mon
02000	DES follows the following rules:
	i) Select an unvisited node s, visit it and treaton
	the current node.
2.71	ii) Find an unvisited neighbor of the current note,
	visit it, and make it the new current node;
	iii) If the current node has no unvisited neighborg.
0	backtrack to the its parent and make that
	the new current node;
	iv) Repeat the above two steps until no more
	nodes can be visited.
	v) If there are still unvisited notes, repeat from
	stept.
	2. Breadth - First Search :
	BFS Fallows the following rules:
	i) Select an unvisited node s, visit it, have it be
	the root in a BFS tree being formed. Its level
	is called the current level.
	ii) From each node & in the current level in the
	order in which the level nodes were visited.
	visit all the unvisited neighbors of m. The
	newly visited nodes from this level form a
	new level that becomes the next
	current level
min	
13	





Date: /20

	(iii) Repeat the previous step until no more nodes
	can be visited.
	iv) If there are still unvisited nodes repeat from
treator	an di tinu i Stepis adminan na distaz (i
	ston inorrus ant
abon In	Conclusion: Thus we have implemented DFS and BFS
	carries want sit silper boo to light
neighbor	STRIVEN ON god stor Janen got at (1)
Foots 3	car brie tarring att att at staintstand
	Labor transmis war actif
9.1000	of little 19952 our evodo and tonggo (vi)
	bedieved and seton
F	er valed batterian links and grants of (v
	tgala
	2. Breadth - Fret Search:
	: 20ther pamollos soft sweller 298
91 1, 901	1. di diesa 2 opour padissipur up dostes ()
vol 21I	the rack in a AFS tree being formed
	is called the current level.
oft of	aval transport at a short con the correct leve
6 017274	larder in which the level nodes were
2017 .11	No 240 Andisa & grizinas and No tizin
p maron	Lovel wind again bading plant toric level
	new art seemed that becomes the new
	Javal Jaamus
	от при
	Cedar

```
Name- Shantanu Deshpande
Class-SYCSE
Roll no.-7
Batch-S1
Input-
#include<stdio.h>
int g[10][10], visited[10];
void bfs(int,int);
void bft(int);
void dfs(int,int);
void dft(int);
void main()
int i,j,n;
char ch='y';
printf("Enter the total no of vertices");
scanf("%d",&n);
printf("\nEnter the edges");
do
do
printf("\nEnter first vertex:");
scanf("%d",&i);
}while(i>n);
do
{
printf("Enter second vertex:");
scanf("%d",&j);
}while(j>n);
g[i][j]=1;
g[j][i]=1;
printf("Want to enter another edge(y/n)");
scanf("%c",&ch);scanf("%c",&ch);
}while(ch=='y'||ch=='Y');
printf("\n\nThe entered graph is");
for(i=1;i \le n;i++)
for (j=1;j \le n;j++)
if(g[i][j]==1)
printf("\nThe edge is between %d and %d ",i,j);
printf("\n\n\nThe breadth first Search is: 1");
bft(n);
for(i=1;i \le n;i++)
visited[i]=0;
printf("\n\n\nThe depth first Search is: ");
dft(n);
}
```

```
void bft(int n)
int i;
for(i=1;i<=n;i++)
if(visited[i]==0)
bfs(i,n);
void bfs(int v,int n)
int w,front=0,rear=-1,Q[10];
visited[v]=1;
while(1)
for(w=1;w<=n;w++)
if(g[v][w])
if(visited[w]==0)
Q[++rear]=w;
visited[w]=1;
printf(" %d",w);
if(rear<front)</pre>
return;
v=Q[front++];
void dft(int n)
int i;
for(i=1;i<=n;i++)
if(visited[i]==0)
dfs(i,n);
}
void dfs(int v,int n)
int w;
visited[v]=1;
printf(" %d",v);
for(w=1;w<=n;w++)
if(g[v][w])
if(visited[w]==0)
dfs(w,n);
```

