	Experiment No: 2: Depth First Search
	Aim: mallations and silens the
	Aim: To implement the Depth - First seasch (DFS)
	algorithm in Python, which traverses or searches
	Harrigh a graph or tree data structure by
	exploring as far as possible along each branch
	before backtracking. is available 85
	Charles and Cherch referred referred)
	AlgerPlam:
1)	Initialize the Graph: Represent the graph using an
	adjacency list or matrix. Each node has a list of
((,, 8	adjacent modes est es connected to
((12)8	Define the DFS Function acate a secussive function
	DFS (graph, node, visited) that explores each node
	and its neighbours and site states
	Mosse the assent mode as visited to avoid
	Revisiting.
	For each adjacent mode that hasn't been visited
	Alcussively apply the DFS function.
3)	Handle unvisited Nodes To the graph is disconnected,
	ensure the DFS function is called for any unvisited
	nodes by "iterating through all nodes in the graph.
4)	Implementation variants: (1) brogge Evillage
	· Recussive DFS: Simple and degant, making use of the call
((1)	stack node : but (input " Fotos 184 stock in stacks
136an_1	· It exative DFS: Using an explicit stack to avoid deep
	Recursive "28 us.
5)	User Input Take input for the number of modes and
	edges, as well as the connections between them.
	THE REPORT OF THE PERSON OF TH
	The second secon

Pagg	am	
------	----	--

Readisive DES implementation:

deg DES (graph, node; visited):

og node not in visited:

print (node, end=" ")
visited. add(node)

for neighbour in graph [mode]:

DFS (graph, neighbour, visited)

Main code 11 + 1000 and

ig _name_2== "_main_2": ~ +20 proofes

nodes: int (input l'Enter the number of nodes: "))
edges: int (input l'Enter the number of edges: "))

experiment were profit proxit seems

Enitialize the graph as a dectionary graph = [i: [] for i in sange (modes)}

hating # Input the edges where I marghe was the

print ("Enter the edges (moder mode 2):")

posse for 2 in sange (edges):

u, V= map lint, Enput 1) spect (1)

edges, as well as the connections behavior to

graph [u]. append (v)

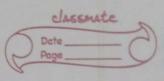
graph [v]. append (u)

start-node = int (input (" Enter the starting mode: "))

print MOFS Traversal starting from node", start-node;

visited = set()

DES (graph start node, visited)



	output: put solar page 1	VISSE.
	The state of the control of the state of the	
	Input:	
Pes		
at a	Enter the number of edges: 7	
1	tilles the eages (mode 1 mode 2):	
85	रहाद रेतवे तार्थात कारायरहरू मिला मिर अन्याहरू रेति	
pai	Penotu estra a sestes o episations sur	
	emptying and peusing between the 1.88 Jugs.	
	1 4	
	2 4 m.31789pgA	
Spair	2H 3 Sola esperantable Reprogent the state 16	(1
uat es	08 a tuple (p. 14) where or 28 Here and unit Age :	
g water	Enter the starting mode on some	
	En Hee 3 - eiles fug.	
	Include state state with both sugs englished	(8
	DFS TRAVESSAL starting from node o:	(E
	0 1 3 (5 1) 4 10 2 2 20 10 4 201 161	
	(E,10) : puis 2010 - E 2011 1154 -	
	temply the stress fug: (0.4)	S 3103
	(1) Emply tile 3- liter fug. (n. 0)	
13 fb.	tens water from the 4-efter Jug to the	
:	मुख धार्मारी हमार हें स्थानिक की सिंद करिएक हैंड मुख्य	
	(n=min(y,4-0), y-min (y,4-0)).	
P3 173	- Pous water grow the 3- liter flig to the a-	
. 22	Jug until one is empty or the others is of	
	(n+min(y,4-2), y-min(y, 4-21).	
	29/6/2 2(9)/9/3 21 270 220	
1 P	1 Resulting (a.o) state validity ast mage trade	
	Thus, the program for Depth First	
	Seasch is successfully executed and the	a outpu
0.00 20	es verfied de bassance es la state att gr	
	end the g. Ates ing is engl	
	seculian es seund	