Exp. No: 2 or (i) u Depth, First Search, ju mo octobe fi Date: Water Jug swillows path append ((ucas uci)) Arm: To implement depth first search (DFs) to traverse a graph & explore all vertices by visiting as for along each branch as possible before backbacking in waterjuy. :0=1[100] Algorithm: rate append ([uan, o]) Step 01: Stout (SI= 100 (pairs) Stepo2: Initialize an empty stack and a list to keep track q visited nodes Stepo3: Push the stouting node onto stack & mark stepons while the stack is not empty, repeat 5 to 7 Stepos: Pop the top node [from the I stack ? Stepo6: Print or process the popped mode is Stepor: For each adjacent uninisted neighbour of the popped node. d= u[i]-ap 8tepos: Stop

(0-5bb00==10) 10 0== 1 11 Program: from collection import dequeue def solution Pa, b, target): (10,15) bioligo. p a append (10,67) m= \$ 2 ः बेर्याच्याव्य वं नवा वी is solvable = False Path: [] ("oldfired test notables") tuing 9= dequeue () 9. append (co.o) (Linky of of the (d) 40 : 11], Indil) tyle i en [(. . gut of. poplett (3tal) trains) this cours (10 trucing to Colffin (13) in m: month of tried

if u(0)>0 or u(1) ber u(0) 20 or u(1) 40 Si confirme but rappy path. append ((uco), uci)] miA To simplement depth first advice [Type, 1974] of if u [o] == taget or u[i] = e target; orale ? and branch as possible before to if ub] == target: if uliji=0: nat nopla Path. append ([u[o1,0]) stop of Stoat 81= len (path) steps: Initialize on emply start or in in the leap print ("("- pout (i)[i),",", pout (i)[i,]")) pte stepons while the stallown) thought sto expend ([uring]) bon got sit got : 30913 for ap in range (max (9,6)+1): Stepon: For each adjacent unique to sollow of the d=u [i]-ap PEPPED node if c == a or (d == o and d >= o): 1. append ([c,d]) bregnis not) q. afferd (aio]): (topat, d, o) noituber 706 a. append ([0,6]) If not is solvouble: substantial si Print (" solution not possible ") [] ital if -- name -- = '-- main - 's) bright P Jug 1 = Port (Poput ("Enter the coundrity of Jug")) Suga: int (input ("Enter capacity of fuga: ")) jug3 = Pnt (input ("Enter Re taget amount")) print (" path grown in Fial state to Solution state")

Solution (jug1, jug2, taget)

S SOLAT o to 1

i miA

: mHiroplA

\$100 or 810st

Output:

Enter the capacity of fug1: 4 Enter the capacity of jugo: 3) Enter the toaget ! 2 -10 1000 of 1000

Path from mitial state to solution state.

(0,0) (1.3) (3,3) 60.37 (4,0) (4,2)

(413) (012) step or limitalise an empty stack to (0,8) or

gall the villed nodes Result: work also short gratuite and down 10 pole

repeat step 5 b7

thus the waterjug program is executed and output is verified. It is it is the

SUPOS: Rep the top node from the stack step of first or process the pepped rock supore to each adjacent weather neighbour show Lagger

84Pos . Marc the neighbour as vicited 86,009. Ruh the unnotted notglipus ants the Step

Stop to Peppeth until all herdrable hader are

人对对外