cercate a DFS program to solve the weater Jug peroblem using python code.

ALGORITHM:

4 liters of weat 1. Intialize the Queue:

I couate a queue q' foer BFS -> breate a set respired to keep track of sixther states to avoid cycles.

-> Enqueue the intial state (0,0) where the both Jugs are empty.

2. BFS loop: Something is not empty.

* Dequew the front state (X, Y) where X is the amount of weater in 3mg 1 and y is the amount of weater in Jug 2.

* If exther x = = tanget on y = = tanget then

* If the state X, y has been visited before Solution is found skip At the next iteration.

* Mark the state (x, y) as resited.

* For the current estate (X, Y) exerciate all passible next states by applying.

· fill Jug 1: (5/19/10 Y)

fill Jug 2: (Jug 2, x)

" Empty Jug 1: (0, Y)

· Empty Jug 2: (x,0)

· Power weather from Jug 1 to Jug 2: - with capacity of Jug 2.

· Power water from Juge to Jug 1: -) with capacity of Jug 1.

> If the Queen is exhausted and the tanget has been enached, perint "solution is not passible" 3: Theck for solution? > otherise, point the sequence of opening leading to the dolution. a show Example: · Tugs has 4 liters of weater Tuge has a liteur of water.

To fill the 2 liteur of water. states to award eyels. con the antid, whate (0,0) nehre; hot Grallons of Grallons of Rule Applied Styn water in 4 water in 3 gallon Jug RI: Intal state on 278.0 (x,9) resident & the R2: Fill3-yallon Jug with & 3 10 1531 04808 1 Rt: Powrall water from 3 to 4 yallon · 3 put 3 Ras Fill 3 yallon Jug 1 2 B5: poros from 3 to4 until fell y Des long referred lestone 2 RT: Empty + yallow Jug PT: Power all wester from 3 to + Jug deep , to the next strention. Demonsteration? ~ (x, x) stoke six shows & words (K'X) straps process of the hold form phylide whomas O athribbed. Ch = 1 (6) : 1 (2) 11 0 31=0 PMC HE=0 P34=3 1 04 LE=3 (K.0) : For per grang. · Sme Dies : (x) 2 put et 10 31=3 tomper de 31=2 Row Reation from Jug 2 to Jug 19 The the capacity of Ing 1. 31=0 462

Kode: C= MOJ -AD from collection imposed dequeue def solution (a, b, target): is solvable = False g. append ([a, 0]) Path=[] q. append (10, 6]) 9 = digne () il mot insolicable; 9. append (60,0)) Pennt C'selution not while len Eq. 70; I - rune == - mais - 1: (Chipat to plue 9: bon left () ([" scot to per continue to I have] has = Court Nontinue NEIJ7b on NEOJ20 on NEIJ<0 Continue solution donte"] path. append ([[u[o], u[i]]) & M[0] == target ou M[i] == target: is solvable = teme & U[0] == target: Extent the colorestal et : Que : Citin to path. append ([u[o], 0]) of uroj!=0: (ro, uroj) Pash from instal asters SI=len (path) for ? in Mange (SI): Print ("(", path [1] [0], "3", Path [1][]")") break. T. append ([UIO], 6]) 9. append ([UII], a]) for an in gange (max (a, b)+1); 1 C= M[O] +ap d= M[i]-ap y c== a con (d==0 and d>=0); Thus the new (Ic, dI) append (Ic, dI) was at well and output is receifed touccessfully.

(Now reflection support they do + EII M = 2 4 (c== 0 and c>= 0) and == bis to be q. append ([c,d]) g. append ([a,o]) q. append (To, b]) () upilo = p Point (" solution not passible") buyers p if not is solvable: of - name == '-main-':

Jug 1= 2nt (annut ("Enter the capacity of Jug !") Jug 2 = int l'apput l'Enter the lapacity of Jugz, Aauget = ist (input ("Enter the Aauget amount:") Puint (" path from initial state to solution state) Solution (Jug1, Jug2, Anarget). of MIO] == tanget an MII] is Laborepe + some output: 2 . W[0] == xanget: Enter the capacity of Jug! the Enter the capacity of Jug 2: 3 Enter the target amount is 2 Path from intial state to solution state SI= len (path) for ? his sange (SI): (0,0) Perint ("(", path [1] [8], 0,3, path [1] [7]")") (4,0) (4,3) (Ed cEasers) brupps of (3,0) J. append ([MIT], al) (1,3) for my in sange (man carte, E) (4,2) 8 90+ [0] N = 5 (0,2) 8 90- [1] N = 6 Thus the weater Jung program is executed and output is receiped successfully.