lab program 8- Puggle Iterative deepening sensich algorithm Algorithm estep 2 define puzzle mode class.

sach mode represents a estata
of the puzzle. Step 2: Define helple functions get lelank hoeition (state)
get meighbors (mode)
apply - action (state, cution)
point esolution (asolution) Step 3: Septh limited worch function depth dimited soarch (mode, goal etate, depth limit) Step 4: Horatie deepening reach iterative despening search (invtix (vial Steps Moin flyution.

1	(Gille 13 ()
	code
	import ropy
	GIOAL STATE = [[1,2,3], [8,0,+], [7,6,5]]
	MOVEUP, MOVE DOWN, MOVELERT, MOVERIGH
	MOVES-EMOVE UP, MOVE POWIN, MOVE LETT
	def priort punde Cotato):
	def print paggle. Corates:
	print (sucue)
	pount O,
	dep fond-blank (state):
	for i in ocange (3):
	for g in Stange (3):
	if state [1] [5]=0:
	oction t, 5
	def moue (state, direttion):
	i, = find blank (Adle)
	new state = copy dec propy (data)
	if disection == MOVE UP and i>0:
	new_state [i][j], new_state [i-1][j] = new_state [i-1][j], new state[i][j]
	new state [1 - 1] [] mew state (i) []
	clse if direction == MOVE_DOWN amen 0/2: new_state [1+1][]-
	new-state [iti][j], new state [i][j]
	ocetwan, new state

classmate

	classmate.
	classmake
	(Pojr
	dep is goal (570to); outurn state == 6,0A) _ 51A-17
	20 turn state == 6,0A) 5,7ATE
	del depth limited search (state, depth, path).
	des agun mureo securi (see sagur, por me
	;f defth ==0:
	geoteven None
	octurn path
	getween Doth
•	
•	for molle direction en MOVES.
	new state = mo le (state move direction) re new - state must in path . new path = path + [new state]
	The state and in the
,	the med -stale still for the passe
	med fain - pain 1 (thus state)
	HISTO - SUMMED SESSON
	(mew_state, depth-1,
	new paths
	·
	outwen None
<u> </u>	
<u> </u>	det :terative doopening search (mittal
	- Stato):
	depots = 0
_	Milolo Tolue 2
	result = defith timeted_ search
	result = defith_dimited_ Search (imittal State, defith, (imittal state))
	if occult is not None:
	. outwon result
`	depth t = 1
	a comment of the comm
4	I had clock - [[] and come of the
	initial state = [[1,4,3], [0,8, 4],[7,65]

(Page / 5 if solution path is not Nome:
print (" solution Found: 1) else: print (" No solution gound ") Initial Stale: [1, 2, 3] [0,8,4] [7, 6, 5]solution found! step 2: [0,8,4-] [7,6,5] [1,2,3]