EX.N0:8

RECURSIVE BEST-FIRST SEARCH ALGORITHM

DATE:24/04/2024

AIM:

To implement the Recursive Best-First Search using python

ALGORITHM:

Step 1: Start with an initial node and add it to a priority queue.

Step 2: While the priority queue is not empty:

Step 3: Pop the node with the lowest estimated cost.

Step 4: If the popped node is a goal state, you've found the solution.

Step 5: Generate child nodes and estimate their costs.

Step 6: Sort child nodes by their estimated costs.

Step 7: Recursively apply RBFS to the child with the lowest estimated cost.

Step 8: If a node returns a failure, update its cost to be higher than the next best child.

Step 9: Continue until a solution is found or all nodes are explored.

Step 10: RBFS terminates when a solution is found or when all nodes have been explored.

PROGRAM:

class Node:

def \_\_init\_\_(self, state, parent=None, cost=0, heuristic=0):

self.state = state

self.parent = parent

self.cost = cost

self.heuristic = heuristic

self.f = cost + heuristic

def is\_goal(self, goal):

return self.state == goal

def generate\_successors(self, goal):

successors = []

if self.state < goal:

successors.append(Node(self.state + 1, self, self.cost + 1, heuristic(self.state + 1, goal)))

return successors

def heuristic(state, goal):

return abs(goal - state)

def rbfs(node, f\_limit, goal):

if node.is\_goal(goal):

return node

successors = node.generate\_successors(goal)

if not successors:

return None

while True:

successors.sort(key=lambda x: x.f)

best = successors[0]

if best.f > f\_limit:

return None

if len(successors) > 1:

alternative = successors[1].f

else:

alternative = float('inf')

result = rbfs(best, min(f\_limit, alternative), goal)

if result is not None:

return result

initial\_state = 0

goal\_state = 5

initial\_node = Node(initial\_state, None, 0, heuristic(initial\_state, goal\_state))

solution = rbfs(initial\_node, float('inf'), goal\_state)

if solution is not None:

path = []

while solution is not None:

path.append(solution.state)

solution = solution.parent

path.reverse()

print("RBFS Path:", path)

else:

print("No solution found.")

OUTPUT:

RBFS Path: [0, 1, 2, 3, 4, 5]

RESULT: Thus the experiment to solve Recursive Best First search algorithm by using python has been executed and verified Successfully.