Artificial Intelligence

Quiz 1

Name: Ahmed Kasteer

Roll Number: 20F-0336

Section: 6C

Code:

import heapq  
  
def astar(start\_node, goal\_node, heuristic\_func):  
 # Initialize the start node  
 start\_node.g\_score = 0  
 start\_node.f\_score = heuristic\_func(start\_node, goal\_node)  
 open\_set = [start\_node]  
  
 while open\_set:  
 # Select the node with the lowest cost  
 current\_node = heapq.heappop(open\_set)  
  
 # Check if the goal node has been reached  
 if current\_node == goal\_node:  
 path = []  
 while current\_node:  
 path.append(current\_node)  
 current\_node = current\_node.parent  
 return path[::-1]  
  
 #checking the childs of the node and neighbours  
 for neighbor in current\_node.neighbors:  
 tentative\_g\_score = current\_node.g\_score + neighbor.cost  
 if tentative\_g\_score < neighbor.g\_score:  
 neighbor.parent = current\_node  
 neighbor.g\_score = tentative\_g\_score  
 neighbor.f\_score = tentative\_g\_score + heuristic\_func(neighbor, goal\_node)  
 if neighbor not in open\_set:  
 heapq.heappush(open\_set, neighbor)  
  
class Node:  
 def \_\_init\_\_(self, name):  
 self.name = name  
 self.neighbors = []  
 self.parent = None  
 self.g\_score = float('inf')  
 self.f\_score = float('inf')  
  
 def add\_neighbor(self, neighbor, cost):  
 self.neighbors.append((neighbor, cost))  
  
def heuristic(node, goal\_node):  
 # this is heuristic function.   
 return ((node.x - goal\_node.x)  
  
# Create the graph  
start\_node = Node('A')  
node\_b = Node('B')  
node\_c = Node('C')  
node\_d = Node('D')  
node\_e = Node('E')  
goal\_node = Node('G')  
  
start\_node.add\_neighbor(node\_b, 4)  
start\_node.add\_neighbor(node\_c, 2)  
node\_b.add\_neighbor(node\_d, 5)  
node\_c.add\_neighbor(node\_d, 8)  
node\_c.add\_neighbor(node\_e, 10)  
node\_d.add\_neighbor(goal\_node, 0)  
node\_e.add\_neighbor(goal\_node, 0)  
  
# Calling the A\* function  
path = astar('A', 'B', '3')  
if path is None:  
 print("No path found")  
else:  
 for node in path:  
 print(node.name)