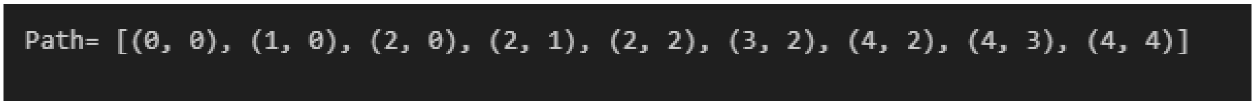
Esha Asif
034
BSAI-3A
Task 7
AI LAB

A* algorithm:

```
#class Node:
#    def __init__(self, position=None, parent=None):
#        self.position = position
#        self.parent = parent

#        self.g = 0
#        self.h = 0
#        self.f = 0
#    def __eq__(self, other):
#          return self.position == other.position

# def a_star_algorithm(maze, start, end):
#    start_node = Node(start, None)
#    end_node = Node(end, None)

#    open_list = []
#    closed_list = []

#    open_list.append(start_node)

#    while open_list:
#        current_node = open_list[0]
#        current_index = 0
#        for index, node in enumerate(open_list):
#            if node.f < current_node.f:
#                current_node = node
#                current_index = index

#        open_list.pop(current_index)
#        closed_list.append(current_node)

#        if current_node == end_node:

#            path = []
#            current = current_node
#           while current is not None:
#               path.append(current.position)
#               current = current.parent
#           return path[::-1]

#       children = []
#       for new_position in [(0, -1), (0, 1), (-1, 0), (1, 0)]:
#           node_position = (current_node.position[0] + new_position[0],
#                            current_node.position[1] + new_position[1])

#           if node_position[0] > (len(maze) - 1) or node_position[0] < 0 or \
#              node_position[1] > (len(maze[len(maze)-1]) - 1) or node_position[1] < 0:
#               continue

#           if maze[node_position[0]][node_position[1]] != 0:
#               continue

#           new_node = Node(node_position, current_node)
#           children.append(new_node)

#       for child in children:

#           if child in closed_list:
#               continue
#           child.g = current_node.g + 1
#           child.h = ((child.position[0] - end_node.position[0]) ** 2) + \
#                     ((child.position[1] - end_node.position[1]) ** 2)
#           child.f = child.g + child.h

#           if child in open_list:
#               existing_child = open_list[open_list.index(child)]
#               if child.g > existing_child.g:
#                   continue
#           open_list.append(child)

# maze = [
#     [0, 1, 0, 0, 0],
#     [0, 1, 0, 1, 0],
#     [0, 0, 0, 1, 0],
#     [0, 1, 0, 1, 0],
#     [0, 0, 0, 0, 0]]

# start = (0, 0)
# end = (4, 4)
# path = a_star_algorithm(maze, start, end)
# print("Path=", path)
```



```
Path= [(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (3, 2), (4, 2), (4, 3), (4, 4)]
```