**PROGRAM:**

from fastapi import FastAPI

from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

app.add\_middleware(

CORSMiddleware,

allow\_origins=["\*"],

allow\_credentials=True,

allow\_methods=["\*"],

allow\_headers=["\*"],

)

def h(p1, p2):

x1, y1 = p1

x2, y2 = p2

return abs(x1 - x2) + abs(y1 - y2)

def box\_adjacent(p, dx, dy):

n = list()

for i, j in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

x = p[0] + i

y = p[1] + j

if x < 0 or x > dx - 1 or y < 0 or y > dy - 1:

continue

n.append((x, y))

return n

def move\_cost(p, barriers):

for b in barriers:

if p == b:

return 10000

return 1

@app.get('/{start}/{end}/{dx}/{dy}/{barriers}')

def search(start, end, dx: int, dy: int, barriers):

start = tuple(map(int, start.split(',')))

**1**

end = tuple(map(int, end.split(',')))

if barriers == '0':

barriers = list()

else:

barriers = list(map(int, barriers.split(',')))

barriers = [tuple(barriers[i:i+2]) for i in range(0, len(barriers), 2)]

g\_cost = dict()

f\_cost = dict()

g\_cost[start] = 0

f\_cost[start] = h(start, end)

done = set()

not\_done = set([start])

parent = dict()

while len(not\_done):

now = None

now\_score = None

for p in not\_done:

if now is None or f\_cost[p] < now\_score:

now = p

now\_score = f\_cost[p]

if now == end:

path = [now]

while now in parent:

now = parent[now]

path.append(now)

# path = path[1:len(path)-1]

for i in barriers:

if (i in path):

return "none"

return path

not\_done.remove(now)

done.add(now)

for adjacent in box\_adjacent(now, dx, dy):

if adjacent in done:

continue

**2**

good\_box = g\_cost[now] + move\_cost(adjacent, barriers)

if adjacent not in not\_done:

not\_done.add(adjacent)

elif good\_box >= g\_cost[adjacent]:

continue

parent[adjacent] = now

g\_cost[adjacent] = good\_box

f\_cost[adjacent] = g\_cost[adjacent] + h(adjacent, end)

return []

from fastapi import FastAPI

from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

origins = [

"http://localhost.tiangolo.com",

"https://localhost.tiangolo.com",

"http://localhost",

"http://localhost:8000",

]

app.add\_middleware(

CORSMiddleware,

allow\_origins=["\*"],

allow\_credentials=True,

allow\_methods=["\*"],

allow\_headers=["\*"],

)

def h(p1, p2):

x1, y1 = p1

x2, y2 = p2

return abs(x1 - x2) + abs(y1 - y2)

def box\_adjacent(p, dx, dy):

n = list()

for i, j in [(1, 0), (-1, 0), (0, 1), (0, -1)]:

x = p[0] + i

y = p[1] + j

if x < 0 or x > dx - 1 or y < 0 or y > dy - 1:

continue

n.append((x, y))

**3**

return n

def move\_cost(p, barriers):

for b in barriers:

if p == b:

return 10000

return 1

@app.get('/{start}/{end}/{dx}/{dy}/{barriers}')

def search(start, end, dx: int, dy: int, barriers):

start = tuple(map(int, start.split(','))

end = tuple(map(int, end.split(',')))

if barriers == '0':

barriers = list()

else:

barriers = list(map(int, barriers.split(',')))

barriers = [tuple(barriers[i:i+2]) for i in range(0, len(barriers), 2)]

g\_cost = dict()

f\_cost = dict()

g\_cost[start] = 0

f\_cost[start] = h(start, end)

done = set()

not\_done = set([start])

parent = dict()

while len(not\_done):

now = None

now\_score = None

for p in not\_done:

if now is None or f\_cost[p] < now\_score:

now = p

now\_score = f\_cost[p]

if now == end:

path = [now]

while now in parent:

now = parent[now]

path.append(now)

# path = path[1:len(path)-1]

**4**

for i in barriers:

if (i in path):

return "none"

return path

not\_done.remove(now)

done.add(now)

for adjacent in box\_adjacent(now, dx, dy):

if adjacent in done:

continue

good\_box = g\_cost[now] + move\_cost(adjacent, barriers)

if adjacent not in not\_done:

not\_done.add(adjacent)

elif good\_box >= g\_cost[adjacent]:

continue

parent[adjacent] = now

g\_cost[adjacent] = good\_box

f\_cost[adjacent] = g\_cost[adjacent] + h(adjacent, end)

return []