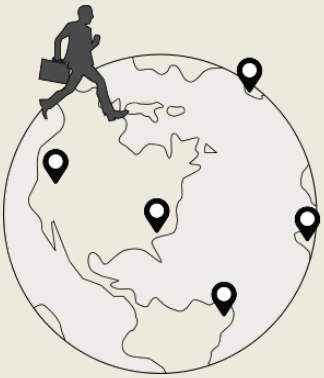
IE_PRACTICA_01	Búsqueda de la ruta más corta entre ciudades (problema del viajante)	IE

Esta práctica consiste en resolver el problema del viajante de negocios, que debe visitar 'n' ciudades una y sólo una vez, buscando la ruta más corta, para invertir menos tiempo y gastar menos combustible.



Para resolver el problema se realiza la optimización del camino más corto utilizando el algoritmo basado en colonias de hormigas.

SOLUCIÓN

Definir las librerías a utilizar

```
import random
import math
```

Función para generar una matriz de distancias de nCiudades x nCiudades

```
# Función para generar una matriz de distancias de nCiudades x nCiudades
def matrizDistancias(nCiud,distanciaMaxima):
    matriz=[[0 for i in range(nCiud)] for j in range(nCiud)]
    for i in range(nCiud):
        for j in range(i):
            matriz[i][j]=int(distanciaMaxima*random.random())
            matriz[j][i]=matriz[i][j]
    return matriz
```

Función para elegir la ciudad a donde ir

```
# Función para elegir un paso de una hormiga, teniendo en cuenta las distancias
# y las feromonas y descartando las ciudades ya visitadas.
def eligeCiudad(dists,ferom,visitadas):
    # Se calcula la tabla de pesos de cada ciudad
    listaPesos=[]
    disponibles=[]
    actual=visitadas[-1]
```

```
    # Influencia de cada valor (alfa:feromonas; beta:distancias)
    alfa=1.0
    beta=0.5
    # El parámetro beta (peso de las distancias) es 0.5, alfa=1.0
    for i in range(len(dists)):
        if i not in visitadas:
            fer=math.pow((1.0+ferom[actual][i]),alfa)
            peso=math.pow(1.0/(dists[actual][i]+0.0000001),beta)*fer
            disponibles.append(i)
            listaPesos.append(peso)

    # Se elige aleatoriamente una de las ciudades disponibles,
    # teniendo en cuenta su peso relativo
    valor=random.random()*sum(listaPesos)
    acumulado=0.0
    i=-1
    while valor>acumulado:
        i+=1
        acumulado+=listaPesos[i]
    return disponibles[i]
```

Función para elegir el camino a seguir por la hormiga

```
# Generar una "hormiga", que elegirá un camino teniendo en cuenta
# las distancias y los rastros de feromonas. Devuelve una tupla
# con el camino y su longitud.
def eligeCamino(distancias,feromonas):
    # La ciudad inicial siempre es la 0
    camino=[0]
    longCamino=0
    # Elegir cada paso según la distancia y las feromonas
    while len(camino)<len(distancias):
        ciudad=eligeCiudad(distancias,feromonas,camino)
        longCamino+=distancias[camino[-1]][ciudad]
        camino.append(ciudad)
    # Para terminar hay que volver a la ciudad de origen (0)
    longCamino+=distancias[camino[-1]][0]
    camino.append(0)

    return (camino,longCamino)
```

Función para actualizar el rastro de feromonas

```
# Función que actualiza la matriz de feromonas siguiendo el camino recibido
def rastroFeromonas(feromonas,camino,dosis):
    for i in range(len(camino)-1):
        feromonas[camino[i]][camino[i+1]]+=dosis
```

Función para aplicar la evaporación de las feromonas

```
# Evapora todas las feromonas multiplicándolas por una constante
# = 0.9 (en otras palabras, el coeficiente de evaporación es 0.1)
def evaporaFeromonas(feromonas):
    for lista in feromonas:
        for i in range(len(lista)):
            lista[i]*=0.9
```

Función "hormigas" para buscar el mejor camino

```
# Resuelve el problema del viajante de comercio mediante el
# algoritmo de la colonia de hormigas. Recibe una matriz de
# distancias y devuelve una tupla con el mejor camino que ha
# obtenido (lista de índices) y su longitud
def hormigas(distancias,iteraciones,distMedia):
    # Primero se crea una matriz de feromonas vacía
    n=len(distancias)
    feromonas=[[0 for i in range(n)] for j in range(n)]
    # El mejor camino y su longitud (inicialmente "infinita")
    mejorCamino=[]
    longMejorCamino=99999999999
    # En cada iteración se genera una hormiga, que elige un camino,
    # y si es mejor que el mejor que teníamos, deja su rastro de
    # feromonas (mayor cuanto más corto sea el camino)
    for iter in range(iteraciones):
        (camino,longCamino)=eligeCamino(distancias,feromonas)
        if longCamino<=longMejorCamino:
            mejorCamino=camino
            longMejorCamino=longCamino
        print("Iteracción(hormiga)",iter," Mejor Camino:", mejorCamino,
"Longitud:",longMejorCamino)
        rastroFeromonas(feromonas,camino,distMedia/longCamino)
        # En cualquier caso, las feromonas se van evaporando
        evaporaFeromonas(feromonas)
    # Se devuelve el mejor camino que se haya encontrado
    return (mejorCamino,longMejorCamino)
```

PROGRAMA PRINCIPAL

Generar una matriz de distancias

```
# Generación de una matriz de distancias
numCiudades=10
distanciaMaxima=100
ciudades=matrizDistancias(numCiudades,distanciaMaxima)
for c in range(numCiudades):
    print("Ciudad",c,":",ciudades[c])
```

Busqueda del camino más corto (en un número de iteraciones)

```
# Obtención del mejor camino
iteraciones=100
distMedia=numCiudades*distanciaMaxima/2
(camino,longCamino)=hormigas(ciudades,iteraciones,distMedia)
print("Camino más corto entre ciudades: ",camino)
print("Longitud del camino más corto: ",longCamino)
```

```
Ciudad 0 : [0, 72, 30, 94, 91, 23, 94, 27, 7, 69]
Ciudad 1 : [72, 0, 96, 57, 24, 0, 23, 56, 90, 46]
Ciudad 2 : [30, 96, 0, 23, 66, 95, 46, 24, 15, 80]
Ciudad 3 : [94, 57, 23, 0, 75, 37, 58, 40, 47, 12]
Ciudad 4 : [91, 24, 66, 75, 0, 3, 30, 48, 26, 92]
Ciudad 5 : [23, 0, 95, 37, 3, 0, 11, 39, 12, 66]
Ciudad 6 : [94, 23, 46, 58, 30, 11, 0, 11, 97, 13]
Ciudad 7 : [27, 56, 24, 40, 48, 39, 11, 0, 66, 90]
Ciudad 8 : [7, 90, 15, 47, 26, 12, 97, 66, 0, 2]
Ciudad 9 : [69, 46, 80, 12, 92, 66, 13, 90, 2, 0]



Iteracción(hormiga) 0  Mejor Camino: [0, 5, 1, 6, 4, 3, 7, 8, 9, 2, 0] Longitud: 369
Iteracción(hormiga) 1  Mejor Camino: [0, 7, 5, 1, 4, 3, 2, 8, 9, 6, 0] Longitud: 312
Iteracción(hormiga) 2  Mejor Camino: [0, 7, 2, 8, 3, 9, 6, 5, 1, 4, 0] Longitud: 264
Iteracción(hormiga) 3  Mejor Camino: [0, 7, 2, 8, 3, 9, 6, 5, 1, 4, 0] Longitud: 264
Iteracción(hormiga) 4  Mejor Camino: [0, 7, 2, 8, 3, 9, 6, 5, 1, 4, 0] Longitud: 264
Iteracción(hormiga) 5  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 6  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 7  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 8  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 9  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 10  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 11  Mejor Camino: [0, 8, 9, 6, 7, 2, 3, 5, 1, 4, 0] Longitud: 232
Iteracción(hormiga) 12  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 13  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 14  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 15  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 16  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 17  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 18  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 19  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 20  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 21  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 22  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 23  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 24  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 25  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 26  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 27  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 28  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 29  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 30  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 31  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 32  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 33  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 34  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 35  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 36  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 37  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 38  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 39  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 40  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 41  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 42  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 43  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 44  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 45  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 46  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 47  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
Iteracción(hormiga) 48  Mejor Camino: [0, 8, 9, 6, 7, 1, 5, 4, 3, 2, 0] Longitud: 220
```

```
Iteracción(hormiga) 49  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 50  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 51  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 52  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 53  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 54  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 55  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 56  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 57  Mejor Camino: [0, 2, 6, 1, 5, 4, 8, 9, 3, 7, 0] Longitud: 209
Iteracción(hormiga) 58  Mejor Camino: [0, 8, 9, 6, 5, 1, 4, 3, 2, 7, 0] Longitud: 206
Iteracción(hormiga) 59  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 60  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 61  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 62  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 63  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 64  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 65  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 66  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 67  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 68  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 69  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 70  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 71  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 72  Mejor Camino: [0, 8, 9, 3, 2, 7, 6, 1, 5, 4, 0] Longitud: 196
Iteracción(hormiga) 73  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 74  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 75  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 76  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 77  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 78  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 79  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 80  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 81  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 82  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 83  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 84  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 85  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 86  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 87  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 88  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 89  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 90  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 91  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 92  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 93  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 94  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 95  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 96  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 97  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 98  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Iteracción(hormiga) 99  Mejor Camino: [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0] Longitud: 166
Camino más corto entre ciudades:  [0, 7, 6, 5, 1, 4, 8, 9, 3, 2, 0]
Longitud del camino más corto:  166
```