



PUCMM

Pontificia Universidad Católica
Madre y Maestra

Facultad de Ciencias de las Ingenierías

Departamento de Ingeniería de Sistemas y Computación

Tarea 2 (T2):

Problema del Viajante

Por:

Ansël E. Corona Coste 2014-0031

Profesor:

J.R Felipe Nuñez

Asignatura:

Diseño y Análisis de Algoritmos

ISC-405-T-001

Santiago de los Caballeros

República Dominicana

Marco teórico:

El problema del Agente Viajero (TSP en inglés) o problema del viajante es una respuesta a la pregunta: *Dada una lista de ciudades y la distancia entre cada par de ellas, ¿Cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad origen?* Este problema es considerado NP-duro dentro de la optimización combinatoria con aplicaciones en la investigación de operaciones y en las ciencias de la computación.

Este problema fue formulado por primera vez en el año 1930 y es uno de los problemas de optimización con mayor auge en la actualidad.

Sus aplicaciones van desde ámbitos simples como la planificación, logística y fabricación de microchips, hasta aplicaciones mas complicadas como el análisis de la secuencia de ADN.

En tanto a la complejidad computacional, la versión de decisión del TSP (donde, dado un largo “L”, la tarea es decidir cuál grafo tiene un camino menor que L) pertenece a la clase de los problemas NP-completos. Por tanto, es probable que en el caso peor el tiempo de ejecución para cualquier algoritmo que resuelva el TSP aumente de forma exponencial con respecto al número de ciudades.

Código:

```
import time

def distancia(punto1, punto2):
    return ((punto1[0] - punto2[0])**2 + (punto1[1] - punto2[1])**2)
    ** 0.5

def distancia_total(puntos):
    return sum([distancia(punto, puntos[index + 1]) for index, punto
in enumerate(puntos[:-1])])

def optimized_tsp(puntos, start=None):
    if start is None:
        start = puntos[0]
    to_visit = puntos
    path = [start]
    to_visit.remove(start)
    while to_visit:
        nearest = min(to_visit, key=lambda x: distancia(path[-1], x))
        path.append(nearest)
        to_visit.remove(nearest)
    return path

def main():
    f = open("ca4663.txt", 'r')
    list = [[float(line.split()[0]), float(line.split()[1])] for line
in f]

    answer = distancia_total(optimized_tsp(list))
    print("The solution is " + str(int(answer)))

if __name__ == "__main__":
    start_time = time.time()
    main()
    print("---- %s seconds ----" % (time.time() - start_time))
```

Conclusión:

```
Music
NetBeansProjects
[Ansel-MacBook-Pro:~ anselcorona$ cd Python
[Ansel-MacBook-Pro:Python anselcorona$ python t2.py
Respuesta: 43451
--- 10.7422149181 seconds ---
Ansel-MacBook-Pro:Python anselcorona$ █
```

La respuesta es mostrada.

Utilicé Python por primera vez para este reporte, me gustó, probablemente envíe los otros en ese mismo lenguaje. (el color scheme del código es porque lo pegué desde Xcode, aunque no se si la tipografía vaya a cambiar entre mi computadora y la suya.

T2✓✓