



Código Asignatura:
ISC-314

Nombre:
Ronald Mariotti

Matricula:
2014-0698

Trabajo:
Tarea 7 - Distancia mínima según Dijkstra

README

En esta tarea lo que se busca es programar la ruta más corta según el algoritmo de Dijkstra que ya conocimos en la clase, para esta tarea hay un archivo con 200 vértices que están desde el 1 al 200. Lo que me pide el problema es encontrar la ruta más corta según el algoritmo de Dijkstra para este grafo, usando el primer vértice como vértice fuente, y para entonces calcular las distancias más cortas entre 1 y todos los demás vértices del grafo. En el orden que se ha pedido, es: 7,37,59,82,99,115,133,165,188,197 se pide presentar la distancia en el mismo orden de las entradas (los diez vértices anteriormente dados), debido a que es lo que se debe tomar en cuenta para cuando se vaya a mostrar el resultado.

NOTAS DE APLICACIÓN:

- Este grafo es lo suficientemente pequeño para que me de $O(mn)$ el tiempo de implementación del algoritmo.

Resultados del programa

Distancias más cortas en orden de las entradas:

7, 37, 59, 82, 99, 115, 133, 165, 188, 197

-

2599, 2610, 2947, 2052, 2367, 2399, 2029, 2442, 2505, 3068

Tiempo = 0.0139999389648 seconds

Captura de resultados

The screenshot shows the Eclipse IDE workspace for a project named 'Workspace Neon - PyDev - Dijkstra/logical/_init_.py'. The main editor displays the 'logical (Dijkstra)' file, which contains the following Python code:

```
1 import time
2 start_time = time.time()
3 def leerGraph(filename):
4     adyacencialist = []
5     lineas = open(filename).read().splitlines()
6     for line in lineas:
7         data = line.split()
8         v = int(data[0])-1
9         for tpl in data[1:]:
10             ts, ws = tpl.split(',')
11             t = int(ts)-1
12             w = int(ws)
13             adyacencialist[v].append((t, w))
14     return adyacencialist
15
16 def extract_min(pq, peso):
17     i = 0
18     j = 1
19     m = peso[pq[0]]
20
21 while True:
22     if j < len(pq):
23         if peso[pq[j]] < m:
24             m = peso[pq[j]]
25             i = j
26     j += 1
27     if j == len(pq):
28         break
29     if i < j:
30         pq[i], pq[j] = pq[j], pq[i]
31         i, j = j, i
32
33 return i
```

The console output shows the execution results:

```
<terminated> _init_.py [C:\Python27\python.exe]
2599, 2610, 2947, 2852, 2367, 2399, 2829, 2442, 2505, 3068
--- 0.0139999389648 seconds ---
```