

Tarea 3 Algoritmos

Planteamiento del problema:

Implementar un programa que lea un CSV con caminos entre ciudades y pueda conectarlos por medio de un gráfico y responder diferentes preguntas asociadas.

Implementación:

En mi implementación, leí el archivo, lo guardé en un objeto "Graph" e implementé el algoritmo Floyd-Warshall para resolver las preguntas.

Estas preguntas son:

1. La ciudad con la menor distancia promedio al resto.
2. Ciudad con el camino más corto a la ciudad X.
3. Ciudades más lejos entre ellas.
4. Ciudades más cerca entre ellas.
5. Listado de las ciudades ordenadas por su distancia promedio al resto.

Resultados:

input_small:

- Ciudad Centrica: Platanares
- Ciudades lejanas: (Boruca, San Pedro)
- Ciudades cercanas: (Platanares, La Virgen)

input_medium:

- Ciudad Centrica: Carmen
- Ciudades lejanas: (San Pedro, Llano Grande)
- Ciudades cercanas: (San José, San Vicente) (Macacona, Pocosol) (Cinco Esquinas, La Trinidad) (Puente de Piedra, La Isabel) (Bahía Ballena, Chirripó) (San Pedro, Santa Rosa) (Pacayas, Picagres) (Jesús, San Isidro) (Venecia, Concepción) (Sarchí Sur, La Cruz) (Laurel, Tacaes) (Chomes, Nicoya) (Nicoya, Daniel Flores) (Monte Verde, San Jerónimo) (Llanuras del Gaspar, San Pedro) (Belén, Pocora) (Santa Ana, Carmen) (San Antonio, Cuajiniquil) (Canalete, Capellades) (Puntarenas, San Rafael) (Río Segundo, Espíritu Santo) (Puerto Jiménez, Laguna)

input_large:

- Ciudad Centrica: Pacuarito
- Ciudades lejanas: (Sarchí Norte, Quitirrisí)
- Ciudades cercanas: (Río Azul, Cipreses) (Bahía Drake, Macacona) (Pacuarito, San Rafael) (San Rafael, Palmira) (Cinco Esquinas, La Trinidad) (La Isabel, Aguas Claras) (Bebedero, San Pedro) (San Jorge, Cutris) (Cutris, Labrador) (Pacayas, Picagres) (La Amistad, Colón) (San Josecito, Palmira) (San Josecito, Tobosi) (Concepción, Sabanillas) (Venecia, Concepción) (Venado, Santa Cruz) (Limón, San Felipe) (San Felipe, Cóbano) (La Ceiba, Santa Bárbara) (Patarrá, Brunka) (Desamparados, La Cruz) (San Pedro,

Monte Verde) (Pavas, Piedras Negras) (Garita, Atenas) (San Pedro, Cariari) (Cariari, Coyolar) (Parrita, Cañas) (Monte Romo, Colima) (Sánchez, Concepción) (Cachí, Santo Domingo) (Santa Cruz, Occidental) (San Rafael, Jacó) (San Rafael, Río Segundo) (San Francisco de Dos Ríos, Damas) (Cangrejal, Tirrases) (Potrero Grande, Tucurrique) (Vuelta de Jorco, Tambor) (Llano Bonito, Jardín) (La Virgen, Cascajal) (Río Naranjo, Granadilla) (San Juan, Cartagena) (Paraíso, Germania) (San Juan de Dios, San Pablo) (Desmonte, Tárcos) (Carrandi, Dulce Nombre de Jesús) (San Rafael Abajo, Cote) (San Juan, Platanares) (Cajón, Yolillal) (Paquera, Quesada) (San Carlos, San Juan Grande) (San Juan Grande, Santa Cruz) (Santa Cruz, Hospital) (Hospital, Tronadora) (Purabá, Roxana) (Arancibia, Merced) (La Cuesta, Guadalupe) (Tures, Sabanilla) (Ángeles, Salitrillos) (Salitrillos, San Roque) (Mercedes, Barrantes) (Buenavista, Corralillo)

Por falta de tiempo tanto para hacer las pruebas y/o preguntar si era necesario, no incluí resultados de encontrar el camino más corto a cada ciudad (opción 2) ni la lista en orden de ciudades por distancia promedio (opción 5).

Manual de uso:

Para compilar, con el Makefile provisto:

1. `make / make release`
 - a. Esto compila y deja el ejecutable en la carpeta `bin/`

Para correr:

Con el Makefile

1. `make run`
 - a. El Makefile tiene una variable `ARGS`, en la cual se ponen los argumentos de línea de comandos, ahí se escribe el path al archivo CSV, por defecto viene el `input_small`
 - b. Para modificarlo, puede hacerse en el Makefile dentro de la carpeta `tp3/` o cuando se hace `make run`
 - i. `make run ARGS=path`

Con el ejecutable

1. `bin/tp3 path`

El Programa tira un menú (en Inglés), es nada más poner los números correspondientes, por alguna razón si se pone un carácter diferente de un número se genera un bucle infinito, pero eso ya es C++ en sí.