

Análisis de tiempos

Por:

Felipe Ochoa

David Ramirez

Datos a analizar:

- Bellman-Ford

- Dijkstra

Universidad EAFIT

Primeramente, vamos a ejecutar el código de Dijkstra con el archivo ciudades.txt y acá podemos ver los resultados del tiempo en segundos.

```
C:\Users\felip\jdk\corretto-17.0.10\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.3\lib\idea_rt.jar=52247:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.3\bin" -Dfile.encoding=UTF-8
Tiempo de ejecución de Dijkstra: 0.001774 segundos
-----Bienvenido a Dijkstra's Algorithm -----
Lista de ciudades con los numeros representando los vertices, en caso de correr el archivo de las ciudades:
Ciudades:
1. Medellin
2. Bello
3. Tunja
4. Quibdo
5. Manizales
6. Yopal
7. San martin
8. Ocaña
```

Tiempo de ejecución de Dijkstra: 0.001774 segundos

Este fue el tiempo ejecutando los caminos mas cortos desde el source 1, que equivale a la ciudad de Medellín.

Ahora observaremos los tiempos de ejecución en el algoritmo de Bellman-Ford para dar una comparación,

```
8. Ocaña
9. Bucaramanga
10. Ibague
11. Cali
12. Neiva
13. Popayan
14. Monteria
15. Mitú
Tiempo de ejecución de Bellman-Ford: 0.000179 segundos
Datos para el source: --> 1

No tiene ciclos negativos
vertices: 15
distancias: 0
infinito: null
```

Tiempo de ejecución de Bellman-Ford: 0.000179 segundos

Este fue el tiempo de ejecución de Bellman-Ford en el mismo archivo de ciudades que el anterior.

De estos datos podemos analizar que los tiempos son un poco similares pero el Bellman-Ford hizo la ejecución en menos tiempo del archivo y en los dos aplicamos que midiera el tiempo de ejecución solo en las funciones de los algoritmos.