

*Escuela de ingeniería y ciencias*

*Unidad de formación TC2038.601*

*Profesor:  
Luis Humberto González Guerra*

## **Evidencia 2**

### **Grupo:601:**

Fernando Morán Fougerat

A01284623

Actividad realizada junto a:

Julen Hoppenstedt Mandiola

A01174098

Fecha de entrega 18 de noviembre del 2024

## Reflexión Individual

Dentro del presente entregable trabajé de manera colaborativa junto con mi compañero Julen para resolver varios problemas dentro de la planeación de agregar infraestructura de cableado a varias colonias. Mi experiencia en el curso de este semestre me ayudó a detectar que los ejercicios donde yo trabajé, el 1 y el 4, podían ser resueltos mediante el uso de Kruskal y de CPP (Closest Pair of Points). Ambos fueron algoritmos que había visto recientemente en clase, que nos permitieron llegar a un resultado de manera más eficiente y con un resultado correcto.

Inicialmente, se nos pidió detectar, la forma óptima de cablear con una nueva fibra óptica conectando colonias de tal forma que se pueda compartir información entre cualesquiera dos colonias en el menor costo posible, aprovechando que ya existen conexiones del nuevo cableado. Esto se pudo lograr como ya mencioné, con un Minimum Spanning Tree ya que podía considerar mi ciudad como un grafo conectado y no dirigido. Este algoritmo tiene complejidad de  $O(E \log E)$ , donde  $E$  es el número de edges, o conexiones que existen entre mis colonias.

Para la siguiente tarea, se nos encargó detectar la ruta más corta posible que visita cada colonia exactamente una vez y al finalizar regresa a la colonia origen, el cual es bastante similar al algoritmo de TSP. Este algoritmo usa Branch and Bound creando nodos con la ruta actual y usando variables para cancelar caminos “podados” que ya no convienen. Este tuvo una complejidad de  $O(2^n)$ . La siguiente respuesta que buscamos fue generar la ruta óptima para ir de todas las centrales entre sí, se puede pasar por una colonia no central. Decidimos usar el algoritmo de Floyd, que usa programación dinámica con 2 matrices para guardar costos mínimos y caminos que usa, con complejidad de  $O(N^3)$ . Ambos problemas los trabajó Julen.

Por último, se nos pidió analizar, para las colonias nuevas dadas, la colonia y punto cartesiano más cercano con el cual se debe conectar (de las ya existentes). Para este problema inicialmente comencé trabajando con el algoritmo de CPP (Closest Pair Point), sin embargo, al darme cuenta que tenía que encontrar el más cercano a uno fijo, encontré un problema. Primero, si seguía la lógica de sortear y luego comparar, tendría que sortear y comparar en base a  $y$ , por lo que siguen habiendo  $O(N)$  comparaciones, más el sorteo de mis colonias  $O(N \log N)$ . Es por esto que decidí seguir un algoritmo de  $O(N)$ , que solo hace una pasada por una colonia nueva.

Esta actividad fue bastante buena para reforzar mis conocimientos sobre algoritmos, estructuras de datos y principios de programación dinámica. Estas tareas me han ayudado a desarrollar competencias clave como lo son Problem solving, y análisis analítico para encontrar la mejor solución con el catálogo de algoritmos que he estudiado y aprendido.