**Actividad Integradora 2**

En esta actividad integradora se cambió el problema que se estaba planteando. Anteriormente la actividad integradora estaba enfocada en la seguridad de las personas al momento de recibir archivos, por lo que se diseñaron algoritmos que permitieran buscar patrones o palíndromos sospechosos dentro de ciertos archivos. En esta ocasión se busca solucionar un problema que tuvo impacto el año pasado en todo el mundo debido a la pandemia. Por lo sucedido, tenemos que crear algoritmos que busquen el camino más corto para instalar el cableado de la fibra óptica. De esta manera se puedan reducir costos e instalar su red de manera óptima apoyándose del uso de algoritmos confiables.

El primer problema que se nos pide solucionar es encontrar la forma más fácil de conectar el cableado, contemplando la cantidad de kilómetros entre distintas colonias de la ciudad, para que se pueda compartir información entre todas éstas. Inicialmente, al analizar el problema se pensó que el utilizar el algoritmo de *dijkstra* sería de gran utilidad para solucionar este problema. Éste busca el camino más corto y funciona considerando el valor de sus vértices; mientras se ejecuta siempre explora todos los caminos más cortos hasta encontrar uno convincente. Lo interesante de este algoritmo es que tiene una complejidad de O(V^2+E), por lo que sí puede servir para resolver el problema si tuviera muchas colonias a contemplar. Además, este algoritmo ya es conocido por ser utilizado dentro de las compañías para resolver este mismo problema que se nos planteó .

En el segundo problema se nos pide buscar la forma de encontrar el algoritmo que encuentre el camino más corto, sin embargo el camino debe pasar por todas las colonias. Esto es porque necesitan repartir volantes de publicidad para que todos los involucrados estén enterados de los informes de las compañías. El algoritmo que se utilizó fue el *Travelling Salesman,* el cual actúa de manera dinámica. El algoritmo busca la distancia más corta posible entre los vértices y después vuelve al inicio; la complejidad del algoritmo es de O(n2^n).

El tercer problema es que la empresa quiere conocer el flujo máximo de información entre las colonias para que los campos electromagnéticos que generan las ciudades no afecten a la red. Por el contexto que tenemos del problema se pensó en un algoritmo que directamente se especializa en encontrar el flujo máximo. En este caso el algoritmo en cada iteración que tiene aumenta el flujo máximo hasta que llega a un punto donde no se puede encontrar un camino que incremente el flujo. Su complejidad es de O(flujo máximo \* E). Es necesario recalcar que este algoritmo es *greedy* debido a que está enfocado a solucionar una problemática en particular. Es por esto que yo lo elegí para esta cuestión.

Para el cuarto problema se contemplarán centrales en distintos puntos de la ciudad y la empresa quisiera saber de qué manera dividir la ciudad para comunicar las casas con cada central y de esta manera saber cuál es la más cercana. Para este problema se concluyó que la manera más adecuada era el algoritmo de *Voronoi*, ya que es la forma matemática para dividir la ciudad y así después encontrar qué casas deberán ser atendidas por las distintas centrales. El algoritmo tiene una complejidad O(n log n) haciéndolo un algoritmo muy valioso para este tipo de problemas.

Solucionar este tipo de problemas se vuelve más eficiente con la ayuda de los algoritmos computacionales, por ello es importante conocer su relevancia. En caso de que una persona, sin conocimientos de esta área, tratara de resolver estos cuatro problemas, le tomaría mucho tiempo para resolverlo y podría realizar algún error que afectaría directamente a la empresa. Ahora, en caso de que se tomaran distintos algoritmos para resolver estos problemas podrían surgir ciertos inconvenientes que van directamente relacionados con la complejidad de los mismos. Al realizar esta actividad integradora y la pasada se tuvo que revisar entre todos los algoritmos posibles cuál era el más conveniente comparando el tiempo de ejecución y obtener la mejor solución posible para cada problema planteado. De esta manera se pudieron encontrar aquellos que funcionan de manera más óptima para los problemas especificados.