E2. Actividad Integradora 2

Emiliano Nuñez Felix A01645413

Sergio Santiago Sánchez Salazar A01645255

2 de noviembre del 2025

Análisis y diseño de algoritmos avanzados (Gpo 604)

Profesor Adolfo Ernesto Arroyo Alanis

Introducción

En esta actividad integradora, se abordó una situación problema compleja que simula la interconexión de colonias mediante fibra óptica, la distribución de recursos, el flujo de datos y la gestión de nuevas contrataciones basadas en ubicación geográfica. Para resolver cada componente de manera eficiente, se seleccionaron e implementaron algoritmos específicos de grafos y geometría computacional. A continuación, se detalla la lógica y el análisis de complejidad de cada solución.

1. Cableado Óptimo: Árbol de Expansión Mínima (MST)

Para conectar todas las colonias con el menor costo posible (fibra óptica), el objetivo es encontrar un subgrafo que conecte todos los vértices sin ciclos y con la suma de pesos mínima.

- Algoritmo utilizado: Algoritmo de Prim
- Explicación: Se eligió Prim sobre Kruskal debido a que trabajamos con una matriz de adyacencia. El algoritmo comienza en un nodo arbitrario (en este caso, el nodo A) y crece iterativamente agregando la arista de menor peso que conecta un nodo visitado con uno no visitado. Para optimizar la selección de la arista mínima, se implementó una cola de prioridad.
- Complejidad computacional: La implementación utiliza un heap binario. En el peor de los casos, procesamos todas las aristas E y vértices V.
    - Complejidad temporal: $O(E \ log \ V)$. Esto es significativamente más eficiente que una búsqueda lineal de la arista mínima en cada paso, la cual sería $O(V^2)$.

2. Ruta de Distribución: Problema del Agente Viajero (TSP)

Se requería encontrar la ruta más corta que visite cada colonia exactamente una vez y regrese al punto de origen.

- Algoritmo utilizado: Held-Karp (Programación Dinámica con Bitmasking).
- Explicación: El problema del TSP es NP-Hard. Una solución de fuerza bruta tendría una complejidad factorial $O(n!)$, lo cual es inviable incluso para grafos medianos. El

algoritmo de Held-Karp optimiza esto dividiendo el problema en subproblemas. Se almacenan los resultados intermedios para evitar recálculos.

- Complejidad computacional:
  - Estados posibles: 2^n (subconjuntos de ciudades) $\times\ n$ (última ciudad visitada).
  - Transiciones: Para cada estado, probamos n posibles ciudades previas.
  - Complejidad temporal: $O(n^2 \cdot 2^n)$. Aunque sigue siendo exponencial, es mucho más rápido que el factorial, permitiendo resolver el problema para un n pequeño de manera casi instantánea.

3. Flujo máximo de datos

El objetivo era determinar la capacidad máxima de transmisión de información entre la colonia inicial y la final, considerando las capacidades de los enlaces.

- Algoritmo utilizado: Edmonds-Karp.
- Explicación: Este algoritmo es una implementación específica del método Ford-Fulkerson que utiliza Búsqueda en Anchura (BFS) para encontrar los caminos de aumento (augmenting paths) en el grafo residual. Al usar BFS, garantizamos encontrar siempre el camino con el menor número de aristas, lo que evita ciclos infinitos y asegura la convergencia.
- Complejidad computacional: En el peor de los casos, cada vez que aumentamos el flujo, al menos una arista se vuelve crítica (saturada).
  - Complejidad temporal: $O(V \cdot E^2)$. Donde V es el número de vértices y E es el número de aristas. Para la escala del problema planteado, esta complejidad es altamente eficiente y asegura la respuesta óptima de flujo.

4. Búsqueda geométrica: Central más cercana

Finalmente, para una nueva contratación ubicada en una coordenada (x, y) arbitraria, se necesitaba determinar cuál de las centrales existentes era la más cercana para asignarle servicio.

- Algoritmo utilizado: Búsqueda Lineal (Vecino Más Cercano / Voronoi).
- Explicación: El problema se reduce a encontrar, dado un conjunto de puntos fijos P, cuál punto p_i minimiza la distancia euclidiana $d(p_i, q)$ respecto al punto de consulta q. Dado

que se trata de una consulta única sobre un conjunto de datos estático, calculamos la distancia desde q a cada una de las n centrales y seleccionamos la mínima.

- Complejidad computacional:
  - Complejidad temporal: $O(N)$. Iteramos una sola vez sobre la lista de centrales, Si bien existen estructuras como k-d-trees o Diagramas de Voronoi que permiten consultas en $O(log\ N)$, su costo de pre-procesamiento no se justifica para una cantidad pequeña de nodos y consultas únicas, haciendo que la búsqueda lineal $O(N)$ sea la solución más práctica y directa.

Conclusión

La combinación de estos algoritmos permitió resolver un problema multifacético de manera óptima. Mientras que Prim y Edmonds-Karp ofrecen soluciones polinomiales eficientes para problemas de grafos clásicos, Held-Karp demuestra cómo la programación dinámica puede hacer tratables problemas intratables (NP-Hard) en escalas pequeñas. Finalmente, el análisis geométrico resalta la importancia de entender el requerimiento exacto (consulta de punto vs. pares más cercanos) para elegir la complejidad adecuada.