

UNIVERSIDAD DE GUADALAJARA
Centro Universitario de Ciencias Exactas e Ingenierías
División de Tecnologías para la Integración Ciber-Humana



Análisis de algoritmos

Jennifer Patricia Valencia Ignacio, Código: 223991721

Elizabeth Arroyo Moreno, Código: 221453749

Karla Rebeca Hernández Elizarrarás, Código: 223991977

Ingeniería en computación

Actividad Voraz

20 de Noviembre de 2025

Introducción

Los algoritmos Prim y Kruskal sirven para resolver cosas como rutas, conexiones y encontrar el costo más bajo. Para probar los algoritmos, hicimos una red de transporte donde cada zona de la ciudad de Guadalajara es un nodo y las distancias entre ellas son aristas con peso.

Estos algoritmos se usan para encontrar la ruta más corta desde un punto en específico hacia todos los demás.

Objetivo

El objetivo es comparar los algoritmos usando Prim y Kruskal para generar el MST y ver si los dos llegan al mismo costo total y luego se analizan los resultados para ver cómo se diferencian los algoritmos en su proceso y en la complejidad que manejan.

Desarrollo

Algoritmo Prim: Nos ayuda a encontrar el árbol de expansión mínima (MST) en un grafo conectado, no dirigido y ponderado. El objetivo de Prim es conectar todo los vértices con el mínimo costo posible sin formar ciclos.

Algoritmo Kruskal: Este algoritmo construye el árbol de expansión mínima (MST). En lugar de ordenar los nodos, usa las aristas.

1. Ordena todas las aristas
2. agrega las aristas más ligeras que no formen ciclos
3. repite hasta que todos los vértices están conectados.

| | Prim | Kruskal |
|---------|---|--|
| Proceso | El árbol tiene un crecimiento desde el vértice inicial de tal manera se selecciona la | Se ordenan las aristas de tal manera que su peso se encuentre de menor a mayor, buscando la arista |

| | | |
|-------------|--|---|
| | arista de menor peso conectándola a un vértice del árbol con uno fuera de, conectando siempre al árbol con una cola de prioridad enfocada en seleccionar las mejores aristas. | menos pesada que no forme ciclos teniendo diferentes componentes desconectados temporalmente. |
| Resultados | En ambos se llega al mismo resultado. | Se podrían generar árboles diferentes pero con el mismo peso total. |
| Complejidad | Con lista de adyacencia $O(E \log V)$ cola de prioridad(HEAP) Con matriz de adyacencia $O(V^2)$ lineal | Con lista de adyacencia $O(E \log V)$ Con matriz de adyacencia $O(E \log V)$ |

Conclusión

Tanto Prim como Kruskal sirven para crear un árbol de expansión mínima (MST), aunque funcionan diferente, los dos algoritmos dieron el mismo resultado para conectar todas las rutas que le dimos.

Prim escoge la arista más barata desde donde empieza, usando una cola de prioridad y Kruskal ordena las aristas de menor a mayor y las va tomando siempre que no formen un ciclo.

Referencias

Levitin, A. (2008). Introduction to design and analysis of algorithms, 2/E. Pearson Education India.

Sedgewick, R. (2013). An introduction to the analysis of algorithms. Pearson Education India.

Wong, K. J. (2024, mayo 10). *Python tree implementation: A guide*. Built In.
<https://builtin.com/articles/tree-python>

Extensión de archivo .bin: todo acerca de la extensión de archivo. (2023, marzo 1). IONOS Digital Guide.
<https://www.ionos.mx/digitalguide/servidores/know-how/extension-de-archivo-bin/>

Prim's algorithm in python. (2024, mayo 28). GeeksforGeeks.

<https://www.geeksforgeeks.org/dsa/prims-algorithm-in-python/>

Kruskal's algorithm in python. (2024, junio 7). GeeksforGeeks.

<https://www.geeksforgeeks.org/python/kruskals-algorithm-in-python/>