UNIVERSIDAD DE GUADALAJARA

Red Universitaria e Institución Benemérita de Jalisco

CUCEI

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

ANALISIS DE ALGORITMOS

Técnica voraz

Algoritmos de Prim y Kruskal

Alumnos:

Prieto Franco César Alejandro

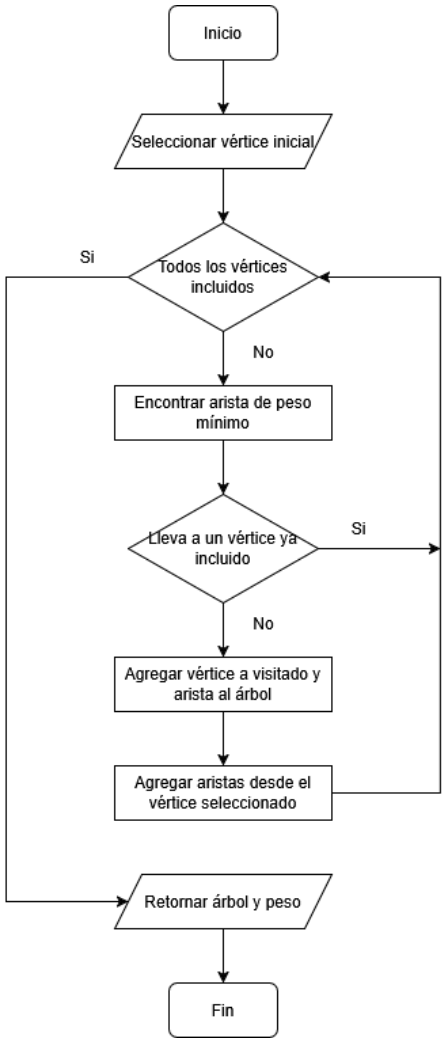
Corona Espinoza Daniel Joel

20 de noviembre 2025

Algoritmo de Prim:

El algoritmo de Prim es un método para construir un árbol de expansión mínima dentro de un grafo ponderado no dirigido. Un árbol de expansión mínima es un subconjunto de las aristas del grafo tales que conecten a todos los nodos sin formar ciclos, y además la suma de los pesos de sus aristas sea la mínima posible que cumpla estas condiciones.

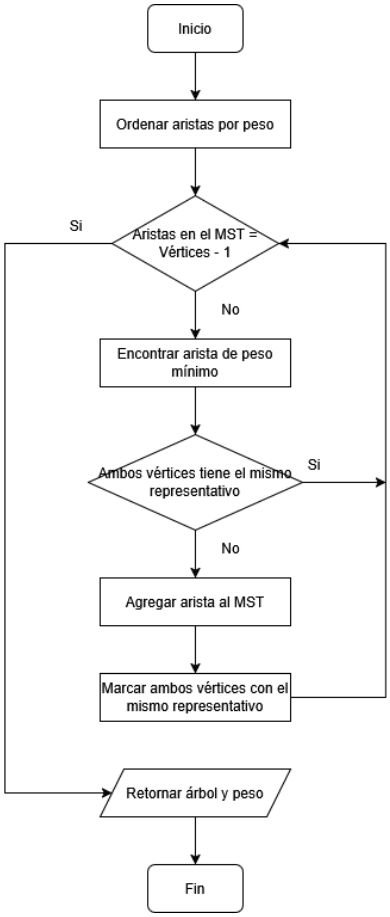
El algoritmo comienza con un árbol que contiene un nodo inicial. Desde este nodo, considera todas las aristas que conectan los nodos del árbol con el resto de nodos del grafo. Luego, toma la arista de menor peso del grafo, verificando que no forme un ciclo que conecte dos vértices que ya se encuentran incluidos en el árbol, y la agrega.



Algoritmo de Kruskal:

El algoritmo de Kruskal, al igual que el algoritmo de Prim, construye un árbol de expansión mínima. Sin embargo, a diferencia del algoritmo de Prim, se basa en conectar las aristas en lugar de los vértices. Para esto, utiliza una estructura de datos de conjuntos disjuntos, en la cual cada elemento pertenece a un conjunto independiente del resto. Para cada uno de estos conjuntos, existe un elemento representativo que comparten todos sus

elementos.
Utilizando una cola de prioridad, se ordenan las aristas del grafo de menor a mayor peso. Por cada una de las aristas, se verifica si los vértices que conecta tienen el mismo elemento representativo. Si es el mismo, la arista forma un ciclo, entonces se descarta. De otra manera, se agrega la arista al MST y se marcan ambos vértices con el mismo representativo, lo que indica que ya están unidos en el mismo conjunto.

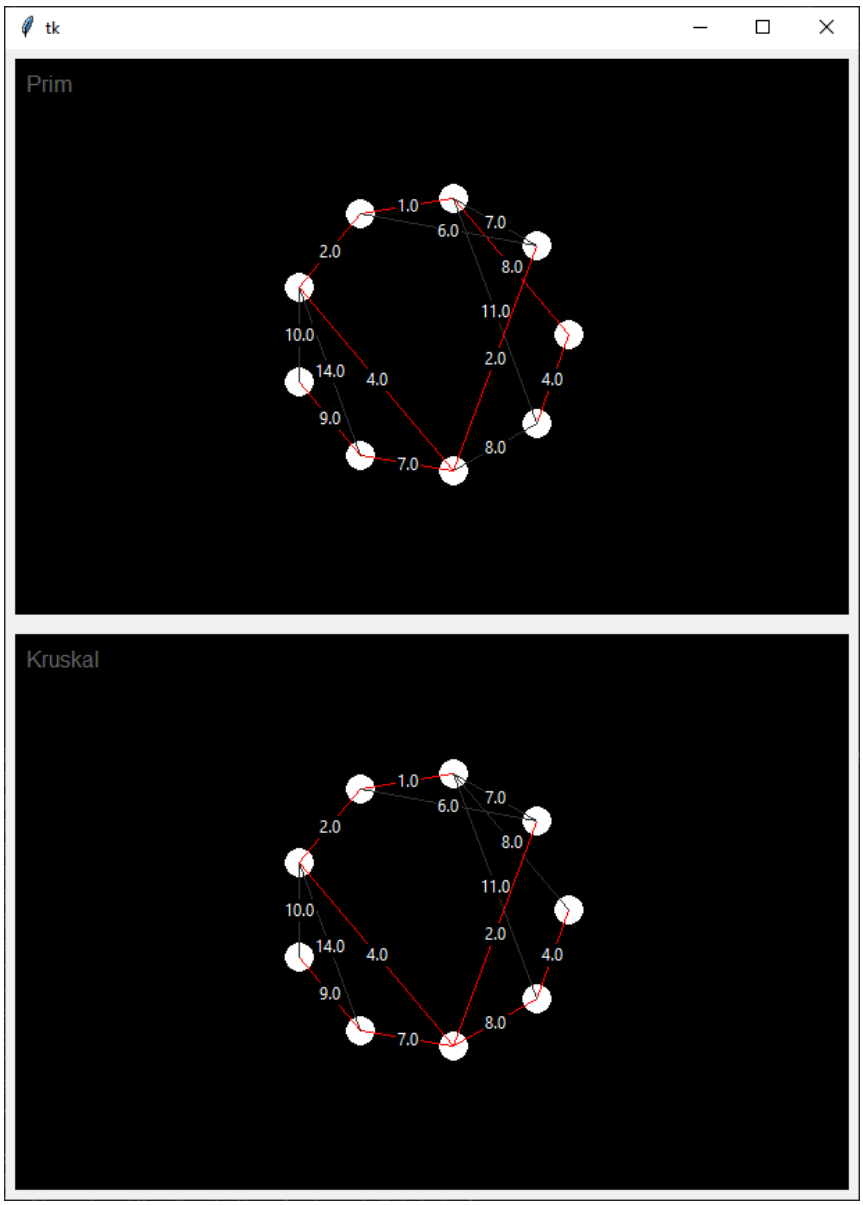


Diferencias en el proceso

Para Prim, se introduce un vértice desde el cuál iniciar a generar el MST. En cambio en el de Kruskal, no se elige un vértice, sólo se inicia a conectar nodos a través de las aristas de menor peso.

Diferencias en resultados

Generalmente, ambos algoritmos generan el mismo MST. Sin embargo, para grafos donde existe más de un MST válido, es posible que cada algoritmo retorne un árbol distinto, aunque del mismo peso.



En esta figura, por ejemplo, Kruskal y Prim difieren por dos aristas, ambas de peso 8. En Prim está presente una, mientras que en Kruskal se puede observar la otra arista.

Complejidad

Kruskal: O(N log N)

Kruskal cuenta con una complejidad O(N log N), donde N es la cantidad de aristas con las que cuenta el grafo, la complejidad se basa en el costo de ordenar las aristas por peso, siendo metodos de ordenamiento N log N lo que le dan la complejidad mayor al algoritmo de Kruskal, consultas y uniones tienen una influencia menor en la complejidad ya que son casi constantes.

Prim: O(N log M)

Prim cuenta con una complejidad de O(N log M) donde N es la cantidad de aristas y M la cantidad de nodos que posee el grafo, siendo el peor caso donde tiene que procesar todas las aristas (N), y hace inserciones y actualizaciones en el heap en tiempo Log M.

Resultados

La implementación de ambos algoritmos fue exitosa, se pudo explorar de manera interesante el comportamiento del algoritmo y cómo se pueden construir con diferentes métodos, como el uso de Heaps para el algoritmo de Prim, o el uso de Disjoint Set Union para el caso de Kruskal. La implementación de ambos sirve para visualizar desde diferentes perspectivas la construcción de árboles de expansión mínima.

Kruskal fue más sencillo de implementar ya que Prim requería del uso de un Heap lo cual volvió más extensa su implementación.

Para experimentar más con la funcionalidad de estos algoritmos se implementó un manera de visualización más dinámica, utilizando librerías de tkinter para poder hacer simulaciones, se realizó una visualización de los gráficos donde utilizará los datos de los nodos para generar pequeñas simulaciones de los grafos

Referencias

GeeksforGeeks. (2025e, agosto 29). Prim's Algorithm for Minimum Spanning Tree (MST). GeeksforGeeks. https://www.geeksforgeeks.org/dsa/prims-minimum-spanning-tree-mst-greedy-algo-5/

GeeksforGeeks. (2025e, agosto 26). Kruskal's Minimum Spanning Tree (MST) Algorithm. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

GeeksforGeeks. (2025c, julio 24). Introduction to Disjoint Set (UnionFind Data Structure). GeeksforGeeks. https://www.geeksforgeeks.org/dsa/introduction-to-disjoint-set-data-structure-or-union-find-algorithm/