

Árbol Parcial mínimo de Prim

¿Qué es?

El algoritmo de Kruskal es un algoritmo de la teoría de grafos para encontrar un árbol recubridor mínimo en un grafo conexo y ponderado. Es decir, busca un subconjunto de aristas que, formando un árbol, incluyen todos los vértices y donde el valor de la suma de todas las aristas del árbol es el mínimo. Si el grafo no es conexo, entonces busca un bosque expandido mínimo (un árbol expandido mínimo para cada componente conexa). Este algoritmo toma su nombre de Joseph Kruskal, quien lo publicó por primera vez en 1956

¿Para qué sirve?

El objetivo del algoritmo de Kruskal es construir un árbol (subgrafo sin ciclos) formado por arcos sucesivamente seleccionados de mínimo peso a partir de un grafo con pesos en los arcos. Un árbol (spanning tree) de un grafo es un subgrafo que contiene todos sus vértices o nodos

¿Cómo se implementa en el mundo?

La implementación del algoritmo de Kruskal para encontrar el árbol de máximo o mínimo costo se puede realizar de la siguiente manera en Python:

Definir la estructura de datos necesaria para representar el grafo. Puede utilizar una lista de aristas, donde cada arista es una tupla que contiene los nodos de origen y destino, así como el costo de la arista.

Implementar la función encontrar y unir para realizar la operación de búsqueda y unión de conjuntos disjuntos. Esto es necesario para evitar la formación de ciclos en el árbol.

Implementar la función kruskal que toma el grafo y devuelve el árbol de máximo o mínimo costo. Esta función utiliza el algoritmo de Kruskal.

¿Cómo lo implementarías en tu vida?

Para determinar que camión me lleva más rápido hacia la escuela o a el trabajo, o si me conviene tomar uno o dos camiones para llegar a tiempo

¿Cómo lo implementarías en tu trabajo o tu trabajo de ensueño?

Lo usaría para determinar cómo mejorar la eficiencia de una maquina o ruta de cierto producto