



Francisco Javier Enriquez Plascencia 21110354

Centro de Enseñanza Técnica Industrial

Inteligencia Artificial

6E

Maestro: Mauricio Cabrera

¿Qué es el árbol parcial mínimo de Prim?

El árbol parcial mínimo de Prim es un algoritmo para encontrar un árbol recubridor mínimo en un grafo conexo, no dirigido y cuyas aristas están etiquetadas. Un árbol recubridor mínimo es un árbol que conecta todos los vértices del grafo y tiene el peso total mínimo posible.

¿Para qué sirve el árbol parcial mínimo de Prim?

El árbol parcial mínimo de Prim se utiliza en una amplia gama de aplicaciones, que incluyen:

- Redes: El árbol parcial mínimo de Prim se utiliza para encontrar la ruta más corta entre dos puntos en una red, como una red de carreteras o una red de telecomunicaciones.
- Distribución: El árbol parcial mínimo de Prim se utiliza para encontrar la ruta más eficiente para distribuir bienes o servicios.
- Construcción: El árbol parcial mínimo de Prim se utiliza para encontrar la ruta más económica para construir una red de infraestructura, como una red de carreteras o una red de tuberías.

¿Cómo se implementa el árbol parcial mínimo de Prim en el mundo?

El árbol parcial mínimo de Prim se implementa en una amplia gama de productos y servicios, que incluyen:

- Navegadores GPS: Los navegadores GPS utilizan el árbol parcial mínimo de Prim para encontrar la ruta más corta entre dos puntos.

- Sistemas de entrega: Los sistemas de entrega utilizan el árbol parcial mínimo de Prim para encontrar la ruta más eficiente para entregar productos.
- Software de diseño de redes: El software de diseño de redes utiliza el árbol parcial mínimo de Prim para encontrar la ruta más económica para construir una red de infraestructura.

¿Cómo implementaría el árbol parcial mínimo de Prim en mi vida?

Podría implementar el árbol parcial mínimo de Prim en mi vida para encontrar la ruta más corta entre dos puntos en mi ciudad. Por ejemplo, podría usar el algoritmo para encontrar la ruta más rápida para ir del trabajo a casa o para encontrar la ruta más corta para visitar a un amigo.

¿Cómo implementaría el árbol parcial mínimo de Prim en mi trabajo o en mi trabajo de ensueño?

En mi trabajo actual, podría utilizar el árbol parcial mínimo de Prim para optimizar las rutas de entrega de los productos. Por ejemplo, podría usar el algoritmo para encontrar la ruta más eficiente para entregar un paquete a un cliente.

En mi trabajo de ensueño, podría utilizar el árbol parcial mínimo de Prim para ayudar a las personas a encontrar el camino más corto para acceder a servicios esenciales, como atención médica o educación. Por ejemplo, podría usar el algoritmo para encontrar la ruta más rápida para llegar a un hospital o para encontrar la ruta más corta para llegar a una escuela.

En general, el árbol parcial mínimo de Prim es una herramienta poderosa que se puede utilizar para encontrar la ruta más corta entre dos puntos en una amplia gama de aplicaciones.

Ejemplo de implementación en la vida real

Imagine que vive en una ciudad con un sistema de transporte público. El sistema de transporte público está formado por una red de autobuses, trenes y tranvías.

Para encontrar la ruta más corta entre dos puntos en la ciudad, podría utilizar el árbol parcial mínimo de Prim para encontrar un árbol recubridor mínimo de la red de transporte público. Este árbol recubridor mínimo representaría la ruta más eficiente para viajar entre cualesquiera dos puntos de la ciudad.

El árbol parcial mínimo de Prim podría implementarse utilizando un algoritmo de programación dinámica. El algoritmo comenzaría con un árbol vacío que consta de un solo vértice, que representa el punto de partida. Luego, el algoritmo agregaría un vértice a la vez al árbol, siempre que el vértice agregado no creara un ciclo en el árbol. El vértice agregado sería el vértice que tiene la menor distancia a un vértice ya en el árbol.

El algoritmo continuaría agregando vértices al árbol hasta que todos los vértices de la red de transporte público estuvieran incluidos en el árbol. El árbol resultante sería el árbol recubridor mínimo de la red de transporte público.

Este árbol recubridor mínimo podría utilizarse para encontrar la ruta más corta entre cualesquiera dos puntos de la ciudad. Por ejemplo, si quisiera encontrar la ruta más corta para ir del trabajo a casa, podría utilizar el árbol recubridor mínimo para encontrar la ruta que pasa por la menor cantidad de autobuses, trenes y tranvías.

Código

The screenshot shows the Spyder Python IDE with a file named 'Árbol Parcial mínimo de Prim.py'. The script implements Prim's algorithm to find a minimum spanning tree. The console output shows the steps for constructing the tree, including the selection of edges and the resulting tree structure.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sun Oct 22 14:04:31 2023
4
5 @author: PC
6 """
7
8 import sys
9
10 def prim_mst(graph):
11     num_vertices = len(graph)
12     # Inicializamos un conjunto vacío para almacenar los nodos visitados.
13     visited = [False] * num_vertices
14     # Inicializamos una lista para almacenar los bordes del MST.
15     mst = [None] * num_vertices
16     # Inicializamos la distancia de cada nodo a infinito.
17     key = [sys.maxsize] * num_vertices
18     # Elegimos un nodo de inicio (puedes personalizarlo según tu gráfico).
19     start_node = 0
20     key[start_node] = 0
21
22     for _ in range(num_vertices):
23         # Encuentra el nodo con la distancia mínima no visitado.
24         min_key = sys.maxsize
25         min_index = -1
26         for v in range(num_vertices):
27             if not visited[v] and key[v] < min_key:
28                 min_key = key[v]
29                 min_index = v
30
31         visited[min_index] = True
32
33         # Agregamos el borde a la lista del MST.
34         if mst[min_index] is not None:
35             print(f"Arista: {mst[min_index]} - Peso: {key[min_index]}")
36
37     # Pasos para construir el Árbol Parcial Mínimo de Prim:
38     # Arista: 0-1 - Peso: 2
39     # Arista: 1-2 - Peso: 3
40     # Arista: 1-4 - Peso: 5
41     # Arista: 0-3 - Peso: 6
```

The console output shows the following steps for constructing the tree:

```
Arista: 0-1 - Peso: 2
Arista: 1-2 - Peso: 3
Arista: 1-4 - Peso: 5
Arista: 0-3 - Peso: 6
```

The final output shows the steps for constructing the tree:

```
In [2]: runfile('C:/Users/PC/3D Objects/6° Semestre/IA/2° Parcial/Tareas/Códigos/Árbol Parcial mínimo de Prim.py',
wdir='C:/Users/PC/3D Objects/6° Semestre/IA/2° Parcial/Tareas/Códigos')
Pasos para construir el Árbol Parcial Mínimo de Prim:
Arista: 0-1 - Peso: 2
Arista: 1-2 - Peso: 3
Arista: 1-4 - Peso: 5
Arista: 0-3 - Peso: 6
```

https://github.com/Arkano007/IA_P2-PR4.git

Bibliografía

colaboradores de Wikipedia. (2023). Algoritmo de prim. *Wikipedia, la enciclopedia libre*.

https://es.wikipedia.org/wiki/Algoritmo_de_Prim