Creación y solución de Laberinto utilizando los grafos(Árbol binario)

* el objetivo del programa:

**El objetivo del programa es implementar los grafos que hemos aprendido en clase para crear un laberinto y solucionarlo. Esto se puede hacer creando un grafo de árbol binario donde identificamos la entrada del laberinto en coordenadas (0,0) y salida.**

* cómo vas a organizar tu código y por qué:

**Lo voy a organizar en un método donde creare el laberinto y lo dividiré en crear el laberinto en una pantalla de plt diferente y luego la solución la dibujo en el camino mas corto del laberinto. Empiezo creando el laberinto porque necesito crear el grafo conectando los nodos de una manera especifica para que sea un laberinto.**

**Cuando ya haya creado el grafo puedo identificar el camino**

**La primera parte de crear el laberinto utilizo el algoritmo de Prim:**

**Este algoritmo primero crea todas las adyacencias a los nodos.**

**Luego vamos visitando otros nodos que estén conectados al nodo anterior si ese nodo no esta ya visitado entonces podemos añadir esta adyacencia a nuestro laberinto. Esto lo hace hasta que haya recorrido todos los Puntos.**

**La segunda parte del algoritmo es crear la solución mas corta del LABERINTO utilizamos entonces el Algoritmo Breadth-First Search:**

**Este algoritmo va mirando la adyacencia de cada nodo hasta que llegamos a la salida donde ya mirando hacia a tras del diccionario de la adyacencia encontramos el camino mas corto que junta la entrada y salida.**

El Reto que encontramos es la de crear un grafo y su Solución ya que tenemos que utilizar diferentes algoritmos y tenemos que encontrar el mas optimo.

Las librería que voy a utilizar son:

**Network este modulo lo utilizo para crear los nodos y edges del grafo**

**matplotlib.pyplot este modulo lo utilizo para dibujar en una pantalla el grafo(laberinto creado por grafo y su solucion)**

**random este metodo se utiliza para que los caminos sean distintos siempre**

**collections es un metodo que utilizamos deque para el algoritmo BFS**

**referencias:**

[**https://networkx.org/documentation/stable/reference/index.html**](https://networkx.org/documentation/stable/reference/index.html)

[**https://www.youtube.com/watch?v=uctN47p\_KVk**](https://www.youtube.com/watch?v=uctN47p_KVk)

[**https://www.youtube.com/watch?v=Y37-gB83HKE**](https://www.youtube.com/watch?v=Y37-gB83HKE)

[**https://www.youtube.com/watch?v=sVcB8vUFlmU**](https://www.youtube.com/watch?v=sVcB8vUFlmU)

[**https://www.youtube.com/watch?v=99oC5Kwxm5w**](https://www.youtube.com/watch?v=99oC5Kwxm5w)

[**https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD\_apUM\_07.pdf**](https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD_apUM_07.pdf)

[**https://www.utm.mx/~jjf/ia/EjemploReporte3.pdf**](https://www.utm.mx/~jjf/ia/EjemploReporte3.pdf)