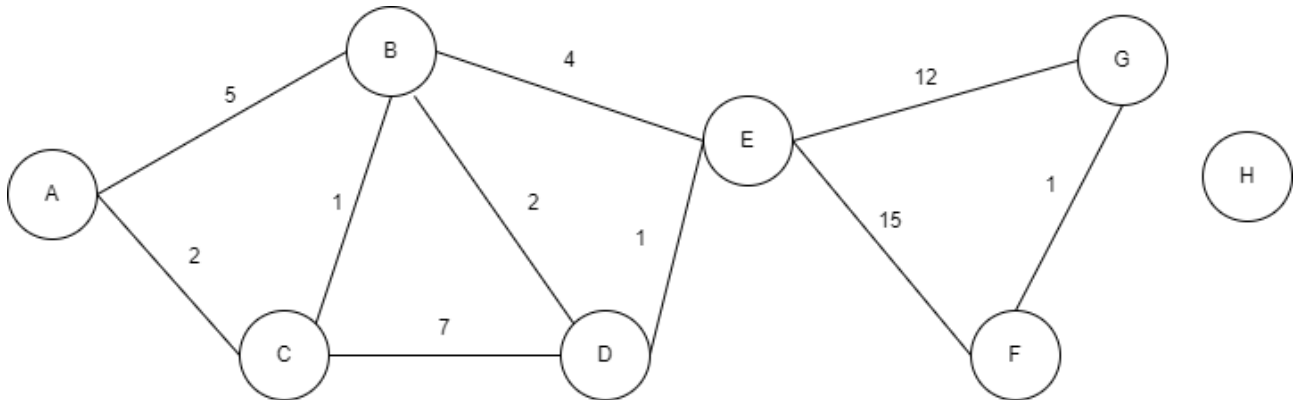


Dado el siguiente grafo, resuelve los siguientes ejercicios:



1. Sigue manualmente los pasos del algoritmo A* para llegar desde A hasta E, rellenando la siguiente tabla:

LISTA ABIERTA	LISTA CERRADA	NODO_SELECCIONADO	NODOS HIJOS

2. Crea un programa en Python para representar el grafo a través de la librería networkX. Elige las coordenadas apropiadas para que el gráfico se asemeje al mostrado arriba.
3. Crea un programa en Python para representar el grafo con las clases Node y Graph del módulo graph.py y codifica:
 - a) Una función que muestre la estructura del grafo con matplotlib*
 - b) Un programa que muestre:
 1. los vecinos del nodo B
 2. Si los nodos G y H están conectados
 3. Si los nodos D y E están conectados
 4. Número de nodos del grafo
 5. Lista de nodos del grafo

* Para mostrar el coste entre nodos puedes situar la etiqueta con el método annotate en el punto medio de las coordenadas de dos nodos:

El punto medio de (x1,y1) y (x2,y2) es (x1+x2)/2, (y1+y2)/2

4. Crea una función reset_graph(g) que borre los rastros de una búsqueda anterior, es decir, que elimine los padres de cada nodo, establezca a -1 su valor heurístico (heuristic_value) e infinito a su coste desde el inicio (distance_from_start).
5. Crea una función ruta(origen,destino,grafo) que utilizando el módulo AStar.py muestre la ruta entre el origen y el destino

6. Realiza las siguientes búsquedas:
 - a) El camino más corto entre el nodo A y el nodo B
 - b) El camino más corto entre el nodo A y el nodo G
 - c) El camino más corto entre el nodo A y el nodo H
 - d) El camino más corto entre el nodo G y el nodo C
7. Realiza las modificaciones necesarias al módulo AStar.py para que, en cada iteración del algoritmo muestre:
 - La listas de nodos abiertos y cerrados
 - El nodo seleccionado
 - Los nodos hijos del nodo seleccionado
8. Realiza la búsqueda entre el nodo A* y el nodo E y compara los resultados del ejercicio 1 y la ejecución del caso A->E en el ejercicio 7. ¿Son iguales? ¿A qué se deben las diferencias?
9. Conversa un poco con ChatGPT de OpenAI o similar y pregúntale:
 - a) Si se considera A* un algoritmo de inteligencia artificial
 - b) Qué algoritmo de búsqueda utiliza Google Maps para encontrar rutas
 - c) Qué alternativas hay al algoritmo A*
 - d) Para qué se usa el algoritmo de Dijkstra
 - e) Para qué se usa el algoritmo de Dijkstra en redes de comunicaciones
 - f) Si se considera Dijkstra un algoritmo de inteligencia artificial
10. Analiza la respuesta a la pregunta f y contesta: ¿Qué razón hay para que A* se considere inteligencia artificial y Dijkstra no?
11. Pídele a la IA que te muestre el algoritmo de A* y compáralo con el que has utilizado en la práctica.
12. Pídele a la IA que te compare el algoritmo A* y el de Dijkstra. Según el chatbot, ¿Cuál es más eficiente?
13. Por último, pregúntale qué otros algoritmos de inteligencia artificial para búsquedas hay y comenta cuál te ha llamado más la atención.