



Primer Examen Práctico - Inteligencia Artificial (Algoritmos de Búsqueda) - (Parejas o Individual)

Instrucciones:

En este examen práctico deberás de resolver un ÚNICO ejercicio que se divide en varios puntos. La situación es la siguiente: Se desea construir un servicio de planeación de rutas aéreas, similar a lo que se hace en páginas como “Google vuelos”, “Despegar.com”, o similares. Para ello, deberás de utilizar algoritmos de búsqueda para explorar las diferentes opciones de rutas que el usuario puede tomar. A continuación, se describen los puntos que debes desarrollar.



Puntos a desarrollar:

Junto con las instrucciones de este examen, encontrarás un archivo llamado **“Ejemplo_vuelos.csv”**, el cual contiene información de ejemplo (del formato) y datos que se tienen de cada uno de los vuelos programados para una fecha en específico, el archivo, contiene los siguientes datos:

- **ID:** Es el identificador del vuelo (Clave única para cada vuelo)
- **Origen:** Es la ciudad desde la cuál despegará el avión
- **Destino:** Es la ciudad a la que llegará el avión
- **Fecha_Salida:** Es la fecha en la cuál el vuelo saldrá desde el origen
- **Hora_Salida_Origen:** La hora exacta a la que despegará el avión de la ciudad de origen, en la fecha de salida
- **Fecha_Llegada:** Es la fecha en la cuál el vuelo llegará al destino (A veces, por las horas de vuelo, podría llegar un día después)
- **Hora_Llegada_Destino:** La hora exacta a la que llegará el avión a la ciudad de destino (En la fecha de llegada)
- **Precio_Vuelo:** Precio total del vuelo (Viaje sencillo)

NOTA: El archivo será SOLO DE EJEMPLO, el día de la presentación del examen práctico, el profesor proporcionará un nuevo archivo con la misma estructura, pero valores diferentes para comprobar el funcionamiento del código.

Con los datos anteriores, deberás desarrollar lo siguiente:

Lectura del archivo y creación de Grafo [3 puntos]:

Primero, deberás de leer el archivo de la manera que consideres más conveniente (se recomienda usar pandas y mostrar los datos como un DataFrame). Y después de eso, deberás crear (de manera automática) el grafo que represente todos los vuelos que haya en el archivo, ten en cuenta que hay algunos que tendrán solo una conexión unidireccional, es decir, que solo habrá vuelos de ida, pero no de regreso.

Deberás de mostrar (de manera gráfica), 2 grafos, el primero de ellos, mostrando el costo EN TIEMPO de tomar cada vuelo, y la segunda, el costo EN DINERO. Ambos grafos se usarán más adelante en el ejercicio.



Algoritmo de búsqueda [4 puntos]:

Esta es la parte más complicada del examen, deberás de crear 4 métodos diferentes:

- Algoritmo de búsqueda por costo uniforme para determinar la ruta más económica EN TIEMPO para llegar de la ciudad "A", a la ciudad "B"
- Algoritmo de búsqueda por costo uniforme para determinar la ruta más económica EN DINERO para llegar de la ciudad "A", a la ciudad "B"
- Algoritmo de búsqueda en AMPLITUD que encuentre TODAS las rutas posibles para llegar a la ciudad "B", desde la ciudad "A"
- Algoritmo de búsqueda en PROFUNDIDAD que encuentre TODAS las rutas posibles para llegar a la ciudad "B", desde la ciudad "A"

Para TODOS los algoritmos anteriores, debes de aplicar las siguientes reglas lógicas:

- Si quieres ir por ejemplo, de la CDMX, a París, y existe un vuelo de la CDMX a Madrid, y luego de Madrid a París, pero el primer vuelo es el 7 de Enero, y el segundo vuelo es el 6 de Enero (un día antes), NO ES POSIBLE tomar los vuelos, por lo que esa ruta NO deberás de considerarla en NINGUNO de los algoritmos. Esto aplica si un vuelo es un día (o más) antes, o si los vuelos son el mismo día, pero la hora del segundo vuelo es anterior a la llegada del primer vuelo.
- Además de lo anterior, deberás de considerar que para que la ruta sea válida, debe de existir AL MENOS una hora de diferencia entre la llegada del primer vuelo y el despegue del segundo vuelo, por lo que si el vuelo "A" llega a las 3PM, y el vuelo "B" sale a las 3:30PM, la ruta A -> B NO ES VÁLIDA, porque el usuario necesitará tiempo para transbordar y cambiar de avión.

Despliegue de la ruta [3 puntos]:

Una vez que tengas todos los a

Rúbrica: En todos los ejercicios, se revisará lo siguiente:

- **Originalidad:** Que el código no sea igual, o sospechosamente similar al de los compañeros del grupo, y que este no haya sido extraído de ChatGPT, o algún servicio similar (Esto es verificable con técnicas de NLP que usará el profesor para evitarlo)
- **Compleitud:** El código debe de contener TODOS los puntos enlistados en cada ejercicio, y la calificación será proporcional a la cantidad de puntos cubiertos del total de solicitados en cada caso.
- **Coherencia:** El código debe de seguir una lógica clara, y NO debe de existir código redundante o sobre escrituras de variables, o incluso código que NO esté haciendo nada y aún así esté presente en la respuesta final.
- **Comentarios:** Todo el código debe de estar debidamente comentado, explicando qué hacen las partes que el alumno considere más importantes en su



código, por ejemplo: *# Entrenamiento del modelo con sklearn*, o *# Uso de input para asignar las variables con las entradas del usuario*

- **Presentación:** Se tomará en cuenta si el alumno agrega markdowns en lugar de comentarios. El cómo muestre los resultados, con prints, imágenes, o cualquier elemento adicional que mejore la presentación del mismo.

Mucha suerte!!!