UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

INTELIGENCIA ARTIFICIAL
"PROBLEMA DE LA RUTA MÁS CORTA"

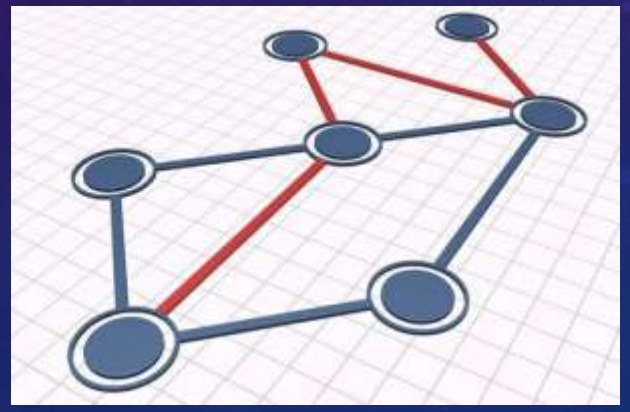
BRENDA SARAHI ORTIZ SOLÍS

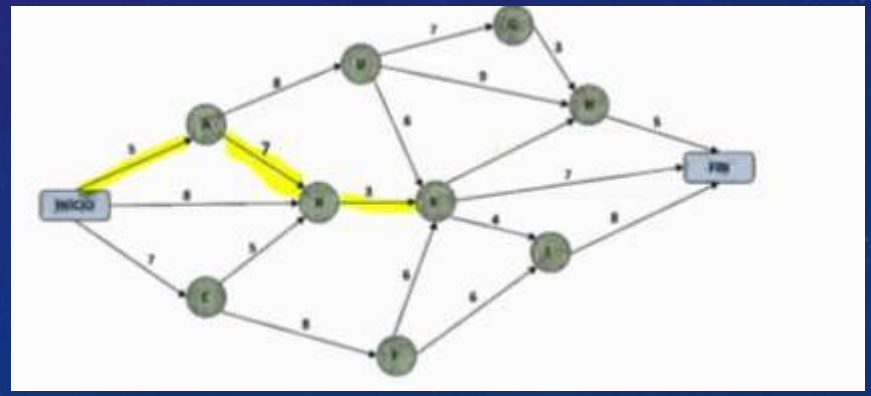
ANDRÉS ALEXIS BRAVO MESTA

FERNANDO ALEXIS ELIZONDO ACEVEDO

DESCRIPCIÓN

- *El problema de resolver es el de la ruta más corta donde tenemos un conjunto de nodos conectados por sus distancias que pueden moverse de un lado a otro según los recorridos que hicimos. Por lo que el objetivo es llegar desde el nodo inicial al nodo final con la ruta más corta*



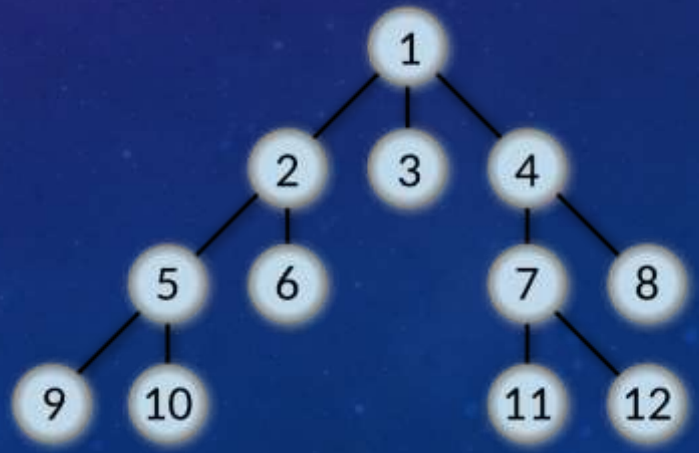


DESCRIPCIÓN DEL PROBLEMA

- Un camionero sale a carretera y empieza cargando gasolina en una estación cercana, pero es un largo camino por recorrer por lo que tendrá que detenerse varias veces en el camino a comprar, cargar mas gasolina o ir al baño por lo que tendrá que decidir en cual de las tantas gasolineras que hay se detendrá y en qué orden para poder llegar con el camino más corto posible

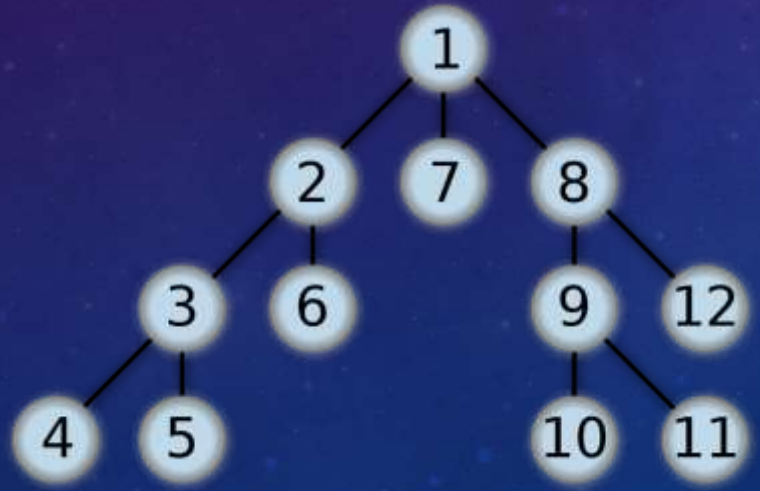
ALGORITMOS UTILIZADOS
1. BFS

- Búsqueda en anchura (en inglés BFS - Breadth First Search) es un algoritmo de búsqueda no informada utilizado para recorrer o buscar elementos en un grafo (usado frecuentemente sobre árboles). Intuitivamente, se comienza en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo) y se exploran todos los vecinos de este nodo. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol.



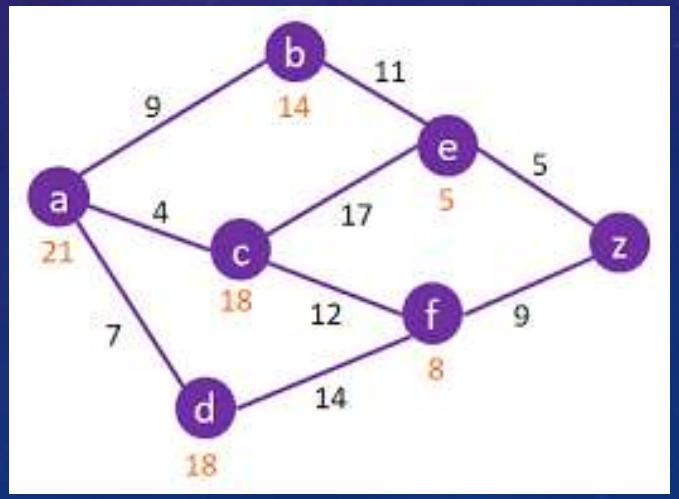
2. DFS

- Es un algoritmo de búsqueda no informada utilizado para recorrer todos los nodos de un grafo o árbol (teoría de grafos) de manera ordenada, pero no uniforme. Su funcionamiento consiste en ir expandiendo todos y cada uno de los nodos que va localizando, de forma recurrente, en un camino concreto. Cuando ya no quedan más nodos que visitar en dicho camino, regresa (Backtracking), de modo que repite el mismo proceso con cada uno de los hermanos del nodo ya procesado.



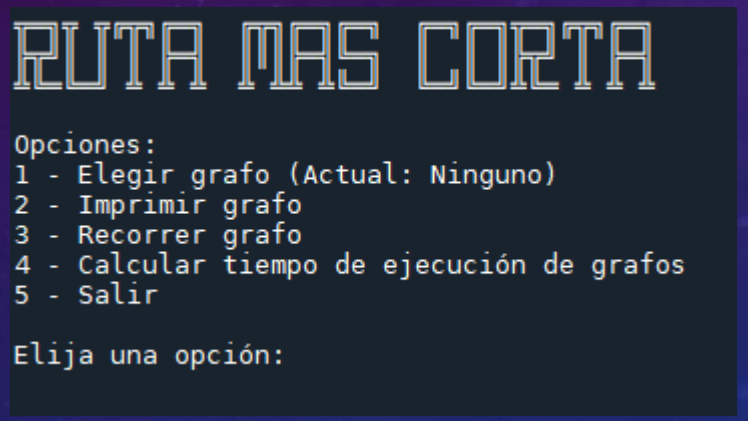
A* STAR

- Este algoritmo es una mejora desarrollada a los postulados del algoritmo Dijsktra que se encarga de encontrar rutas más cortas dentro de un grafo. En esta modificación se toma como punto central la observación búsquedas informadas dentro del grafo que nos permitan tomar decisiones óptimas sobre los caminos que deben tomarse para recorrer de forma eficiente el grafo

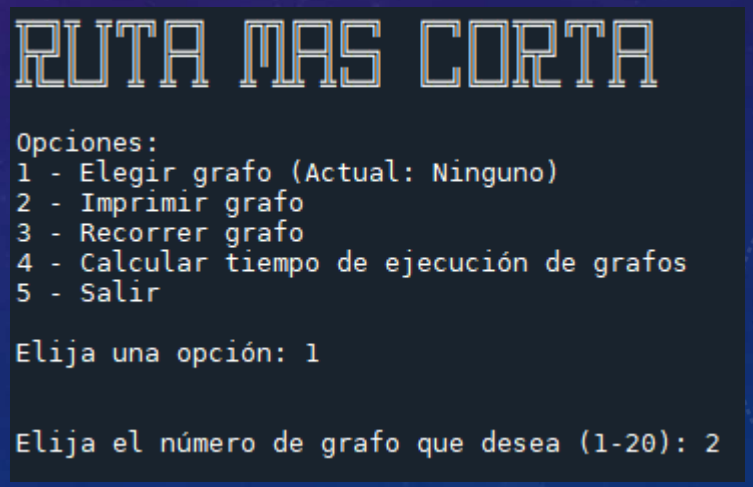


DEMOSTRACIÓN DEL PROGRAMA

Este es el menú principal, primero debemos de elegir el grafo que vamos a utilizar en este ejemplo.
Presionamos 1 para elegir el grafo

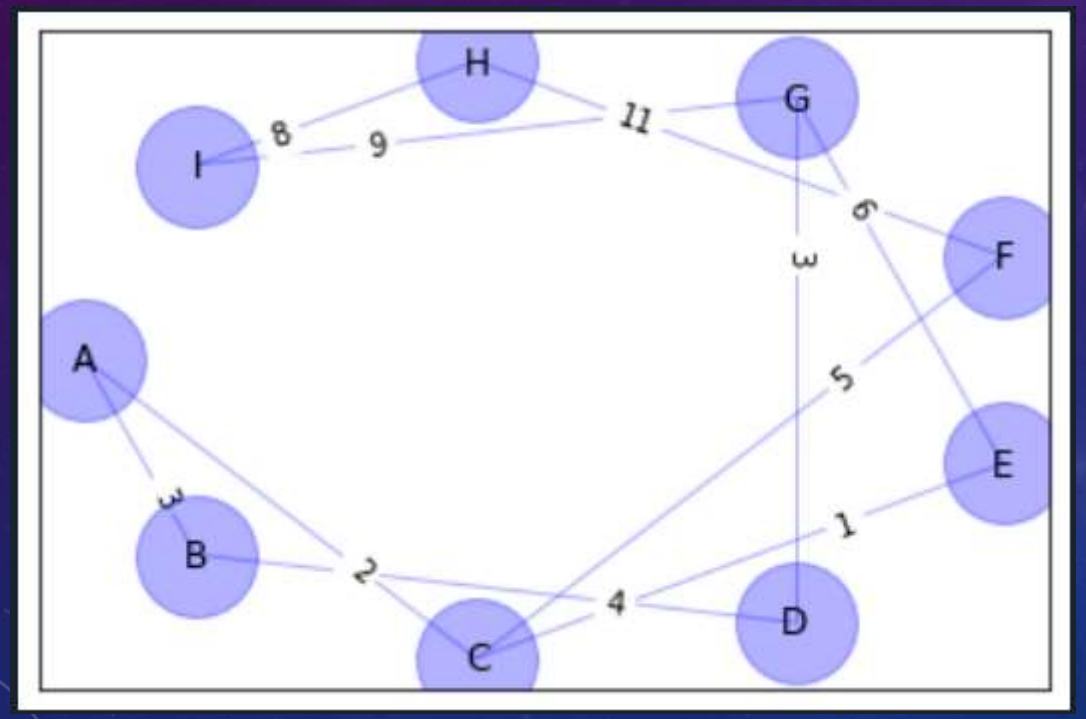


Después nos pedirá que ingresemos un numero del 1 al 20 que son grafos diferentes, nosotros elegiremos el 2

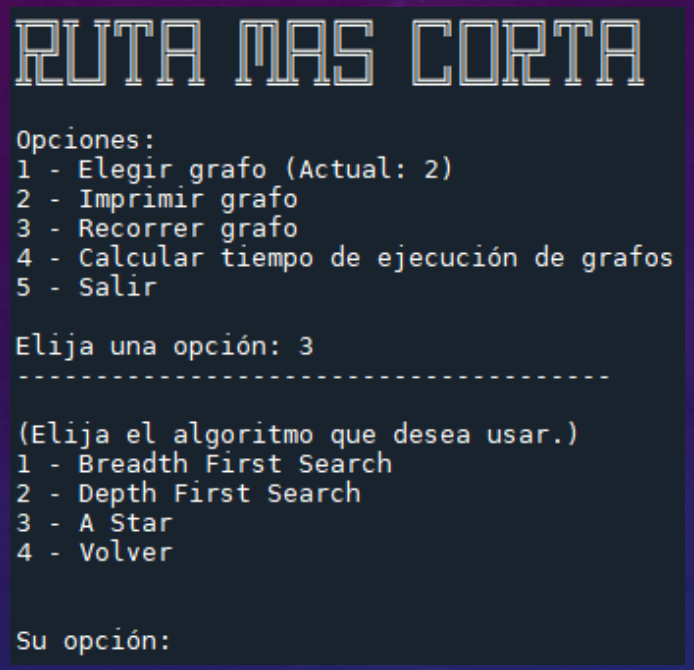


PROBLEMA

Nos regresara al menú principal y ahora presionamos 2(Imprimir grafo) para que nos muestre el grafo que elegimos



- NOTA : Este grafo simula las posibles paradas y distancias entre ellas en el recorrido del camionero



Al ver el grafo seleccionado, nos regresara el menú principal, ahora procedemos a hacer correr el grafo presionando 3. Para después tener que elegir con cual de los 3 algoritmos deseamos hacer el recorrido.

Con cada uno de los algoritmos nos lanzaran los siguientes resultados

RESULTADOS

BFS:

```
[RECORRIDO CON BFS]
A -> B -> C -> D -> E -> F -> G -> H -> I

Nodos recorridos: 9
Costo: 20
Tiempo: 9.91058349609375 microsegundos

(NOTA): El tiempo del algoritmo varía mucho en cada ejecución.
```

DFS:

```
[RECORRIDO CON DFS]
A -> B -> D -> G -> I

Nodos recorridos: 5
Costo: 8
Tiempo: 0.9970664978027344 microsegundos

(NOTA): El tiempo del algoritmo varía mucho en cada ejecución.
```

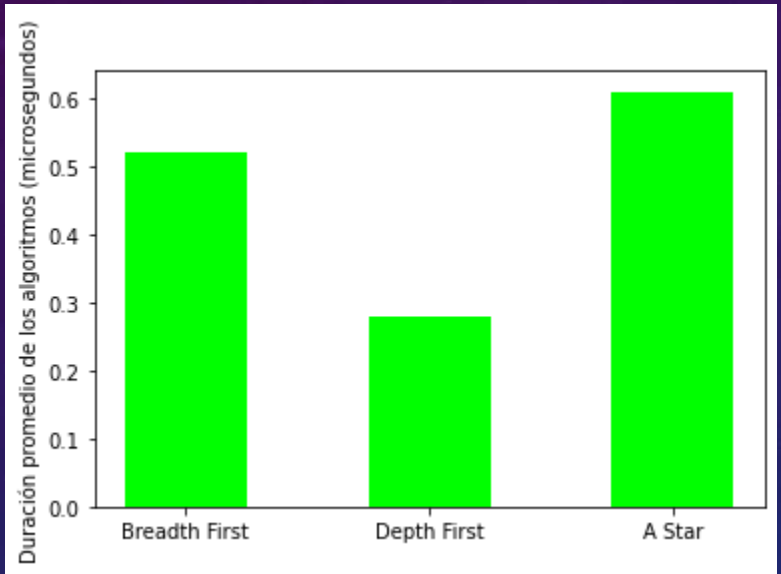
A Star*

```
[RECORRIDO CON A*]
A -> C -> E -> G -> I

Nodos recorridos: 5
Costo: 4
Tiempo: 9.965896606445312 microsegundos

(NOTA): El tiempo del algoritmo varía mucho en cada ejecución.
```

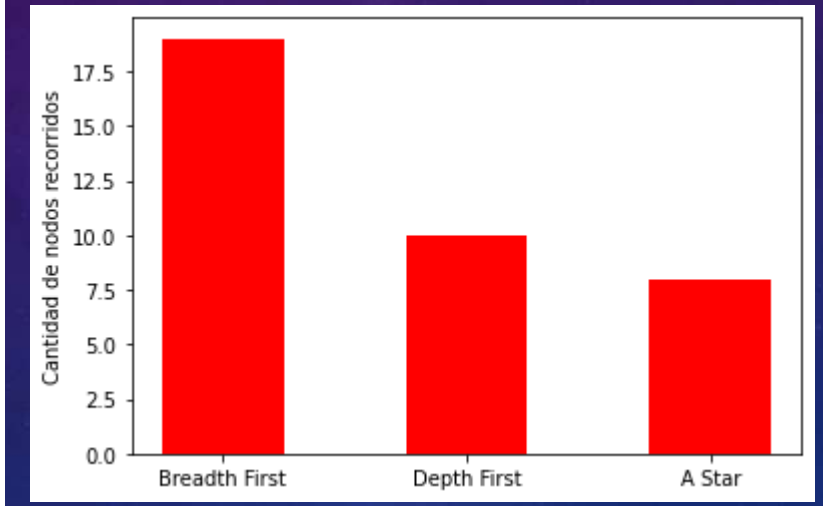
- Al terminar de hacer el recorrido de los grafos presionamos 4 para volver al menú principal, y ya estando ahí para poder ver los datos anteriores pero ahora de manera grafica presionamos 4 (Calcular tiempo de ejecución de grafos)

GRÁFICAMENTE

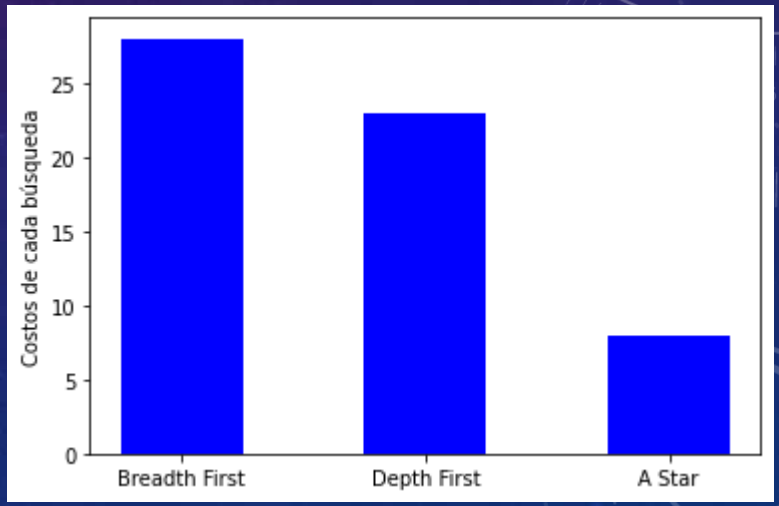
Duración promedio de los logaritmos (microsegundos)



Cantidad de nodos recorridos



Costos de cada búsqueda



CONCLUSIÓN

- Mientras que el A Star es el algoritmo que más tarda de los tres, también es el más óptimo en términos de los nodos recorridos y el coste del viaje. Aunque la búsqueda en profundidad tardó menos tiempo en ejecutarse, generó un costo relativamente alto, y tuvo que recorrer más nodos que el A Star.
- El A Star es de los tres, el algoritmo que obtiene el costo más óptimo. Sin embargo, si lo que se desea es obtener una solución rápida, los otros dos algoritmos son más óptimos que este.