

EL PROBLEMA DEL MONO Y LOS PLÁTANOS

Equipo #3

Descripción de los parámetros

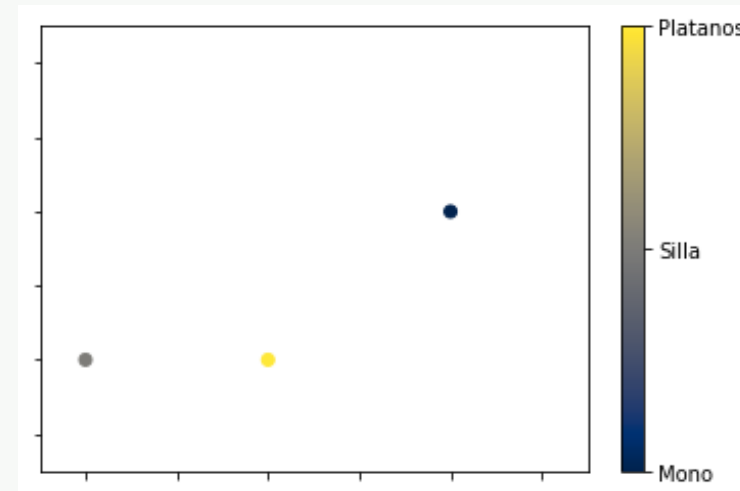
- Los parámetros a considerar para nuestro problema son:
 - ❖ Posición del mono.
 - ❖ Posición de los plátanos.
 - ❖ Posición de la silla.
 - ❖ Tamaño del mundo.
- La posición la consideramos las coordenadas dentro de la cuadrícula con la que se esté trabajando al momento.

Descripción de las reglas del ambiente y del agente

- El mono es nuestro agente en el problema y puede realizar las siguientes acciones:
 - ❖ El mono se puede mover hacia arriba.
 - ❖ El mono se puede mover hacia abajo.
 - ❖ El mono se puede mover hacia la izquierda.
 - ❖ El mono se puede mover hacia la derecha.
 - ❖ El mono se puede mover en diagonal.
 - ❖ El mono puede agarrar un objeto (silla o plátanos).
 - ❖ El mono puede soltar un objeto (silla).
 - ❖ El mono se puede subir a la silla.

Descripción de cómo se representa el ambiente y agente

- La representación del problema se hace por medio del módulo matplotlib de Python. Se generan dos listas donde se guardan las coordenadas x y y del mono, silla y plátanos. Se muestra la gráfica generada por el módulo, donde cada parámetro esta representado por un color diferente, tras haber generado un movimiento, el cuál es calculado a partir de las funciones específicas de los algoritmos y posteriormente almacenado como la nueva posición dentro de las listas.



PSEUDOCÓDIGOS

A*

A*

1. Inicio.
2. Se abre el archivo de texto donde se ingresaron los parámetros y se guardan dichos datos en dos listas(x, y).
3. Se genera una lista bidimensional utilizando el tamaño de la cuadrícula obtenido previamente donde se guardarán los valores obtenidos por la función del algoritmo y se rellena cada posición con el valor None.
4. Se calcula el valor de cada nodo disponible y se guarda este valor en la lista.
5. Se busca el valor mínimo de los nodos disponibles y se regresan sus coordenadas para ser guardadas como las nuevas posiciones del mono.
6. Se agrega el nodo actual a la lista de nodos visitados.
7. Se muestra la cuadrícula.
8. Si el mono se encuentra en la posición destino actual se avanza al siguiente paso. En caso contrario se regresa al paso 4
9. Se eliminan las coordenadas del destino actual de las listas y se despejan las listas que contienen los valores y los nodos visitados.
10. Si aún se tiene un destino se regresa al paso 4.
11. Fin

DFS

DFS

1. Inicio.
2. Se abre el archivo de texto donde se ingresaron los parámetros y se guardan dichos datos en dos listas(x, y).
3. Se genera una lista con los nodos visitados y otra con los nodos cerrados.
4. Se revisa si los nodos vecinos han sido visitados previamente y elige uno que no haya sido visitado como el siguiente movimiento.
5. En caso de que no haya ningún nodo disponible, se regresa al nodo previamente visitado y al paso 4.
6. Se agrega el nodo actual a la lista de nodos visitados.
7. Se muestra la cuadrícula.
8. Si el mono se encuentra en la posición destino actual se avanza al siguiente paso. En caso contrario se regresa al paso 4
9. Se eliminan las coordenadas del destino actual de las listas y se despejan las listas que contienen los valores y los nodos visitados.
10. Si aún se tiene un destino se regresa al paso 4.
11. Fin

HILL CLIMBING

HILL CLIMBING

1. Inicio.
2. Se abre el archivo de texto donde se ingresaron los parámetros y se guardan dichos datos en dos listas(x, y).
3. Se calcula la distancia desde cada nodo vecino hasta el objetivo actual.
4. Se busca el valor mínimo de los valores calculados y se regresan las coordenadas para ser guardadas como las nuevas posiciones del mono.
5. Se muestra la cuadrícula.
6. Si el mono se encuentra en la posición destino actual se avanza al siguiente paso. En caso contrario se regresa al paso 3
7. Se eliminan las coordenadas del destino actual de las listas.
8. Si aún se tiene un destino se regresa al paso 3.
9. Fin

GRÁFICAS COMPARATIVAS

Ejemplo #1

- ❖ Mono: 0, 0
- ❖ Silla: 4, 1
- ❖ Plátanos: 0, 3

A*

Numero de movimientos: 12

Tiempo: 9.213045120239258

DFS

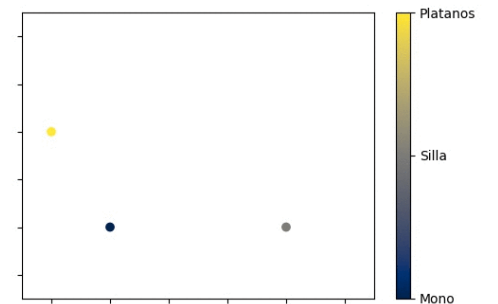
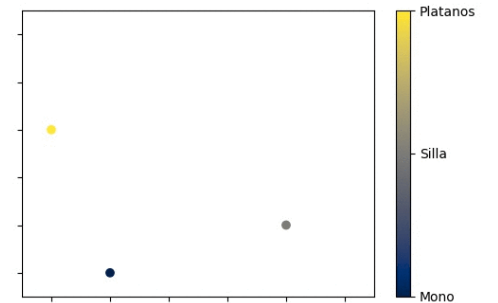
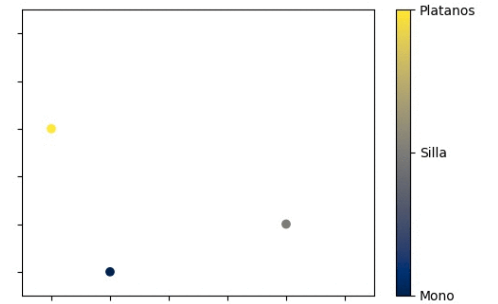
Numero de movimientos: 15

Tiempo: 11.20598816871643

Hill Climbing

Numero de movimientos: 8

Tiempo: 6.595128774642944



Ejemplo #2

- ❖ Mono: 0, 2
- ❖ Silla: 4, 2
- ❖ Plátanos: 1, 1

A*

Numero de movimientos: 7

Tiempo: 5.943850040435791

DFS

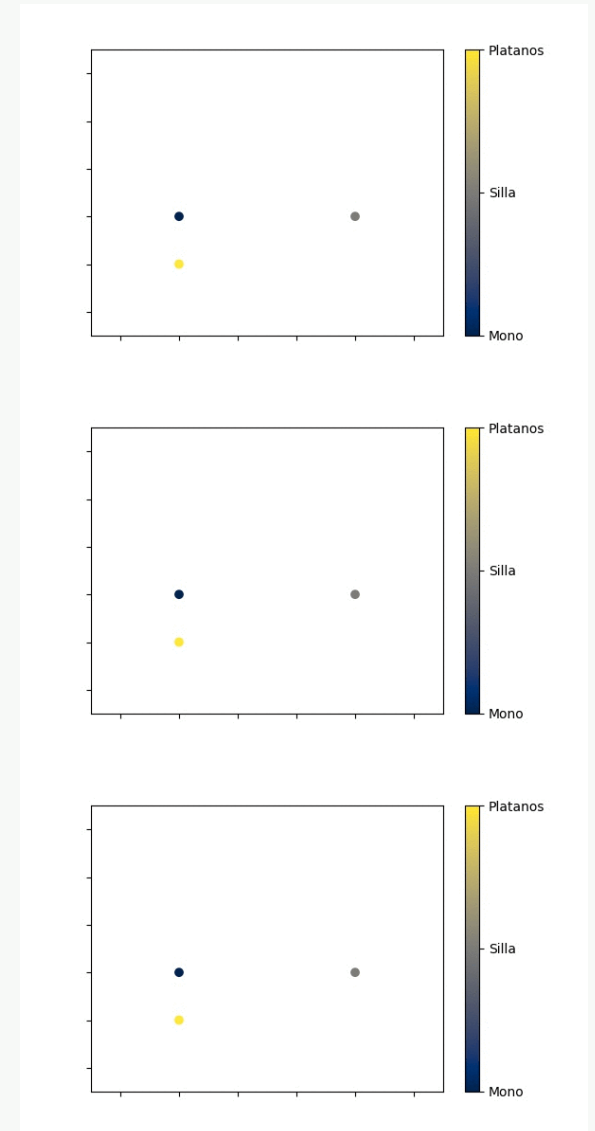
Numero de movimientos: 51

Tiempo: 34.84068155288696

Hill Climbing

Numero de movimientos: 7

Tiempo: 5.8857152462005615



Resultados

Promedios para A*:

Número de movimientos: 7.25 movimientos

Tiempo de ejecución: 6.06265 segundos

Promedios para DFS:

Número de movimientos: 34 movimientos

Tiempo de ejecución: 23.72 segundos

Promedios para Hill Climbing:

Número de movimientos: 4.9 movimientos

Tiempo de ejecución: 4.5849 segundos

Conclusión

En base a las observaciones apreciadas con los resultados, podemos concluir que el algoritmo de Hill Climbing resultó ser el más eficiente tanto en tiempo como en movimientos, seguido del algoritmo A*, dejando en último lugar y por un amplio margen al algoritmo DFS que no resultó ser el más adecuado para nuestro problema propuesto.