

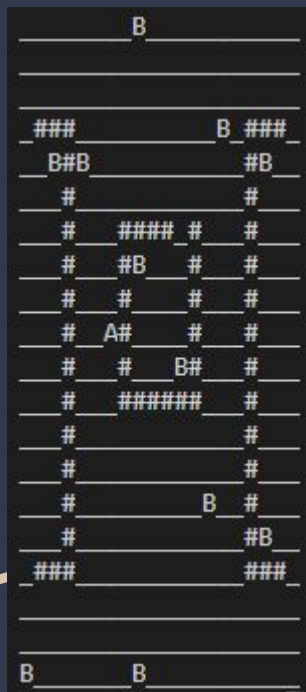
Proyecto Final Inteligencia Artificial.

| | |
|-----------------------------------|---------|
| Elí Israel Delgado Escárcega | 1821213 |
| Francisco Javier Martínez Palomar | 1937837 |
| Kevin Alexis Nájera Abrego | 1798194 |

Mundo de la aspiradora



El problema



Tenemos un **agente aspiradora** posicionado en un mundo de $n \times n$ losetas, y cuyo objetivo es recoger toda la suciedad del mapa.

- Loseta transitable (representada con un símbolo «_»).

- Loseta con muro (representada con un símbolo «#»).

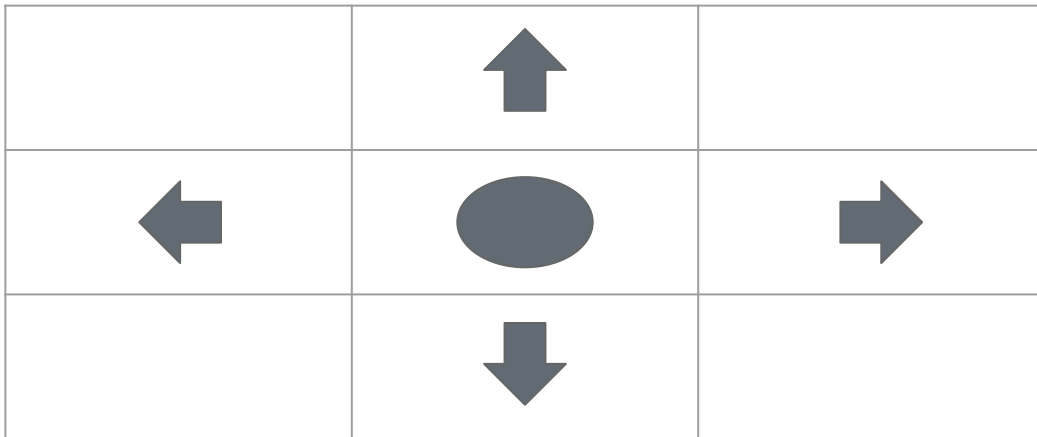
- Loseta con basura (representada con el símbolo «B»).

- Loseta donde se encuentra o expande la aspiradora (representada con el símbolo «A»).

Reglas del mundo

Nuestra aspiradora puede **moverse** en 4 direcciones, norte, sur, este y oeste, con la restricción de que no puede salirse de los límites del mapa, y, además, no está permitido que avance a una loseta en la que se encuentra un muro.

La aspiradora desconoce el tamaño del mapa, pero reconoce cuando llega al límite del mapa.



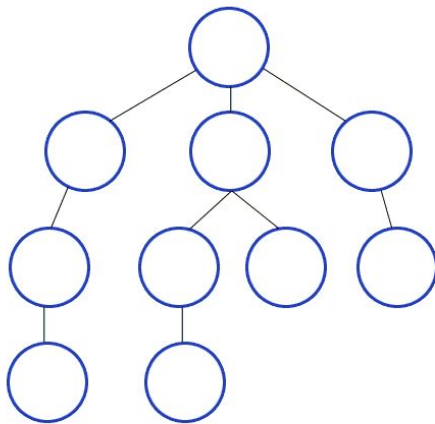
Reglas del mundo

La aspiradora desconoce la cantidad de **basura** que hay en el mapa, su objetivo es recoger toda la basura en el menor número de movimientos. Para ello, hace uso de un algoritmo bfs. Una vez que encuentra una basura, actualiza el mapa e itera el bfs tantas veces como sea necesario hasta encontrar que ya no haya basura.



El algoritmo usado, BFS.

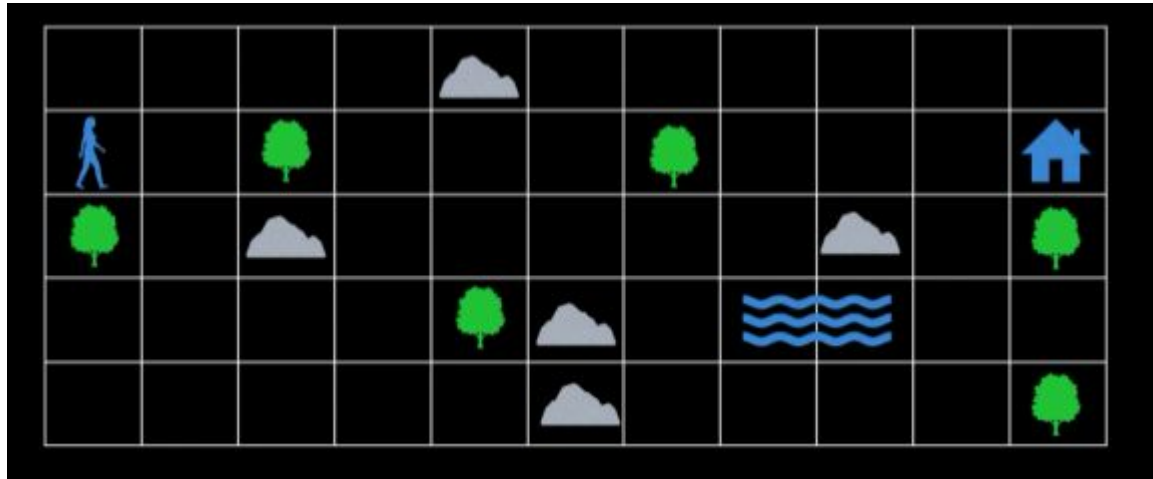
El algoritmo bfs es un algoritmo de búsqueda. Es utilizado para explorar nodos en un grafo en amplitud. Y es particularmente útil para encontrar la ruta más corta en grafos no ponderados.



BFS para una cuadrícula.

Muchos problemas de grafos se pueden representar con una cuadrícula, porque puedes determinar los vecinos de un nodo basado en sus posiciones.

Ir de un punto A a B con obstáculos en medio, es un ejemplo.



BFS para una cuadrícula, gráficamente.



BFS en nuestro algoritmo



Tamaños de las instancias y parámetros.

La idea del programa es que este pueda resolver 20 mapas distintos dados por el programador. Estos 20 mapas son de dimensión $n \times n$, empezando desde 20 filas/columnas y hasta 30 filas/columnas, dos mapas de cada dimensión.

Para que programa pueda resolver el problema se le debe dar un mundo o mapa con las características ya mencionadas. El nodo inicial es necesario, pero este se encuentra dentro del código al iterar hasta encontrar la loseta con el carácter 'A'.



=

A

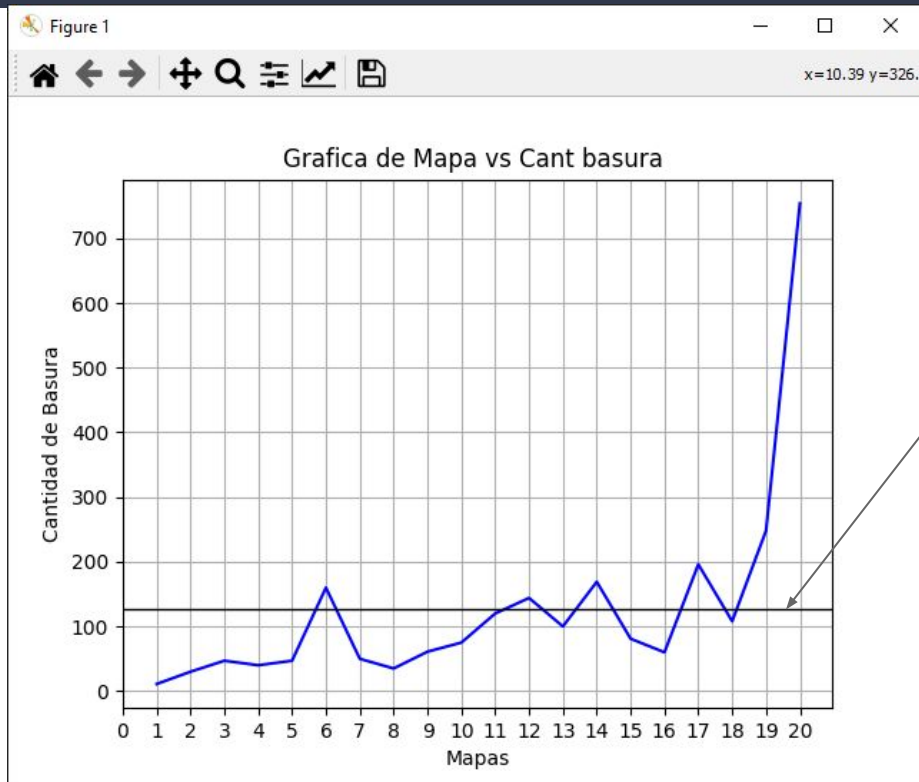
Gráficas

El programa se corrió con 20 instancias de distintos tamaños. A continuación, se muestra la gráfica mapa vs tiempo y mapa vs cantidad de basura a manera de comparación. Nótese que para estas gráficas no se imprimieron los pasos, solo se realizó el algoritmo. Esto último con el objetivo de ver el tiempo real de ejecución.

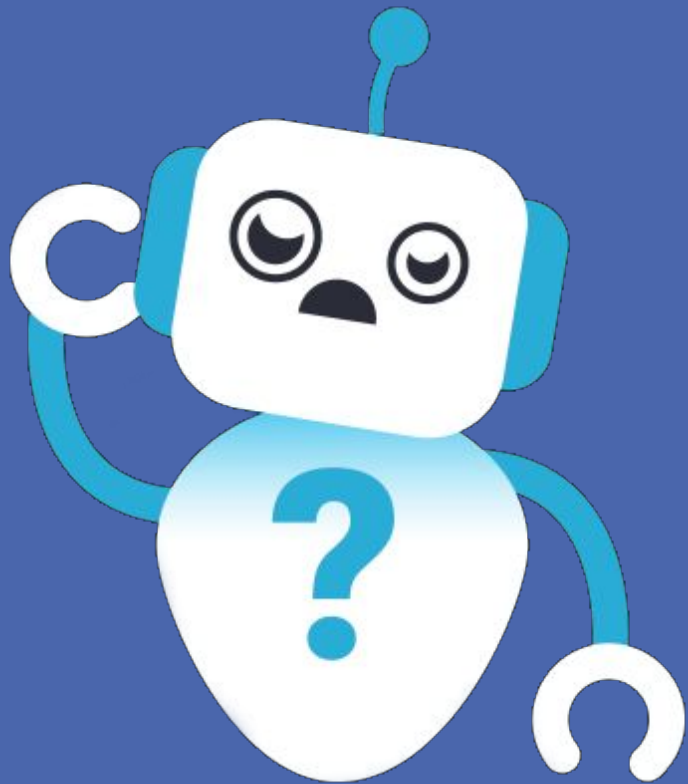
Mapa vs Tiempo



Mapa vs Cantidad de Basura



¿Preguntas?



Referencias de ayuda

Fiset, William. [WilliamFiset]. (2018, abril, 20). Breadth First Search Algorithm | Shortest Path | Graph Theory. [Archivo de video]. [Breadth First Search Algorithm | Shortest Path | Graph Theory - YouTube](#)

Fiset, William. [WilliamFiset]. (2018, abril, 2). Breadth First Search grid shortest path | Graph Theory. [Archivo de video]. [Breadth First Search grid shortest path | Graph Theory - YouTube](#)