

Trabajo Práctico 1

Santiago Libonati, Tomás Maiza

habría que explicar más o menos qué solución dimos y qué resultados obtuvimos

Ejercicio 1

El algoritmo DFS busca los nodos más profundos del árbol de búsqueda sin expandir nodos ya visitados.

Nuestra implementación lleva una pila de direcciones (el camino recorrido), una pila que contiene los nodos próximos a visitar (los sucesores de los nodos que visita) y un conjunto de estados ya visitados.

Siguiendo el algoritmo de búsqueda general, cuando visitamos un nodo chequeamos si es meta, en cuyo caso retornamos la lista de direcciones acumuladas. En caso contrario apilamos sus sucesores no visitados en la pila de nodos a visitar (es decir, los visitamos primero). Cada nodo recorrido es agregado al conjunto de nodos visitados y la dirección tomada para llegar a él es agregada a la pila del path. Cuando retrocedemos en el árbol (porque el camino tomado no nos lleva a nada) desapilamos tantas direcciones como diferencia haya entre el nivel del nodo actual y el tamaño de la pila, ya que el camino tiene exactamente un nodo por cada nivel. Por ejemplo, si llegamos al nivel 4 del árbol, la pila del path podría verse así:

```
path = ['N', 'N', 'E', 'S']
```

Si el siguiente nodo visitado es un nodo de nivel 2, ahora el path será:

```
path = ['N']
```

es decir, borramos el camino recorrido hasta volver al nivel 2, donde será pusheada la dirección tomada para llegar al nodo actual.

Los resultados obtenidos fueron los siguientes:

- Mapa tinyMaze: 16 nodos expandidos, costo 10
- Mapa mediumMaze: 146 nodos expandidos, costo 130
- Mapa bigMaze: 391 nodos expandidos, costo 212

```
tomi@tomi-lenovo:~/Escritorio/IIA/IIA-TP1-Pacman/search$ python3 pacman.py -l tinyMaze -p SearchAgent
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 10 in 0.0 seconds
Search nodes expanded: 16
Pacman emerges victorious! Score: 500

tomi@tomi-lenovo:~/Escritorio/IIA/IIA-TP1-Pacman/search$ python3 pacman.py -l mediumMaze -p SearchAgent
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 130 in 0.0 seconds
Search nodes expanded: 146
Pacman emerges victorious! Score: 380

tomi@tomi-lenovo:~/Escritorio/IIA/IIA-TP1-Pacman/search$ python3 pacman.py -l bigMaze -p SearchAgent
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 212 in 0.1 seconds
Search nodes expanded: 391
Pacman emerges victorious! Score: 298
```

Ejercicio 2

Ejercicio 3

Ejercicio 4

Ejercicio 5

Ejercicio 6

Ejercicio 7