

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos:	
	Nombre:	

Actividad 1: Software y ejemplo de caso de prueba con simple-ai en Python

El software indicado para realizar la práctica es el que se puede instalar a partir de la página siguiente: se trata de la librería en python "simple-ai" (<https://pypi.org/project/simpleai/>).

Script para realizar la experimentación

Se proporciona el script 202324_RYP_actividad1.py que está pensado para ejecutar varias pruebas sobre un mismo laberinto y mostrar los resultados pedidos. Aunque la salida de este script son tres resultados, tenéis que añadir el cuarto, que corresponde a ejecutar A* con la heurística que falta.

El ejemplo que se incluye es sólo para este documento, observad que es diferente del que se pide en la actividad. Este ejemplo es sólo para poder explicar los diferentes campos de la salida

1. BREADTH-FIRST SEARCH (AMPLITUD)

```
#####
#           #           #   #

# #####          #   #
# T #   #           #   #
# . ###          ##### #
# .          #####.... #   #
# .....#.#. #   ##### #
# #####... #.   #.P #
#           #   #..... #

#####
Total length of solution: 34
Total cost of solution: 33.0
max fringe size: 9
visited nodes: 151
iterations: 151
```

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos:	
	Nombre:	

2. DEPTH-FIRST SEARCH (PROFUNDIDAD)

```
#####
#           #           # #
# #####           # #
# T.#   #           # #
#  .###   #####   ##### #
#  .....##### .....#   #
#           ....# .#   .#   ##### #
#   #####.....#   .....# P #
#           #           #   .... #
#####
Total length of solution: 38
Total cost of solution: 37.0
max fringe size: 39
visited nodes: 69
iterations: 69
```

3. A* CON LA HEURÍSTICA PREDEFINIDA

```
#####
#           #           # #
# #####           # #
# T.#   # .....   # #
#  .### .....##### .##### #
#  .....#####   .#           #
#           # #   .#   ##### #
#   #####   #   .....#.P #
#           #           #   ... #
#####
Total length of solution: 34
Total cost of solution: 33.0
max fringe size: 22
visited nodes: 139
iterations: 139
```

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos:	
	Nombre:	

Presentación de resultados

Los resultados anteriores se pueden tabular:

Ejemplo 1	Longitud solución	Coste solución	Nodos visitados
Amplitud	34	33	151
Profundidad	38	37	69
A* con heurística (¿cuál es?)	34	33	139
...etc...			

Comentario sobre resultados en ejemplo1: comparación de longitud, coste, nodos visitados, razones por las que puede ocurrir esto. ¿Son las soluciones óptimas? ¿Esto lo garantiza el algoritmo en cuestión?

Si lo deseáis podéis pintar gráficas comparativas para cada algoritmo y valor capturado. Este tipo de comparativa, muy sintética, es la que podéis hacer para cada situación.

Modificaciones para la práctica

- Para modificar el mapa hay que variar la variable MAP. Se puede cargar desde un fichero si queréis.
- Para cambiar la heurística, tendréis que modificar la función heuristic de la clase GameWalkPuzzle.
- Para cambiar los costes, tendréis que modificarlos en la variable COSTS

Dependencias para el Visor Web

Para utilizar WebViewer, tenéis que instalar **pydot y flask**.