**LAB 1: Search**

**Q1: The Pacman board will show an overlay of the states explored, and the order in which they were explored (brighter red means earlier exploration). Is the exploration order what you would have expected?**

Si, los nodos más brillantes, es decir, los explorados antes, se corresponden a los que están en el inicio del recorrido. A medida que llegamos a nuestro objetivo, el overlay se vuelve más oscuro, es decir, son los nodos que hemos explorado hacia el final. El último nodo explorado es el final.

***Hint:* If you use a Stack as your data structure, the solution found by your DFS algorithm for mediumMaze should have a length of 130 (provided you push successors onto the fringe in the order provided by getSuccessors; you might get 246 if you push them in the reverse order). Is this a least cost solution? If not, think about what depth-first search is doing wrong.**

En mediumMaze obtenemos un coste de 130.

No es la solución menos costosa, pues no tenemos en cuenta los costes de cada movimiento y en el peor de los casos recorreriamos todos los nodos para encontrar el nodo final.

**Does Pacman actually go to all the explored squares on his way to the goal?**

No, pacman no recorre todos los nodos explorados hasta llegar al final ,ya que utiliza las acciones directas para llegar al nodo final proporcionadas por DFS.

**Q2: Does BFS find a least cost solution? If not, check your implementation.**

**DFS:** Coste en mediumMaze= 130 , Nodos Expandidos = 146

**BFS:** Coste en mediumMaze = 68 , Nodos Expandidos = 269

DFS es más costoso pese a expandir menos nodos que BFS.

**Q4: You should see that A\* finds the optimal solution slightly faster than uniform cost search (about 549 vs. 620 search nodes expanded in our implementation, but ties in priority may make your numbers differ slightly). What happens on openMaze for the various search strategies?**

**bigMaze:**

Coste A\* : 210 Nodos Expandidos A\* : 549

Coste UCS : 210 Nodos Expandidos UCS : 620

**openMaze:**

Coste DFS : 298 Nodos Expandidos DFS : 806

Coste BFS : 54 Nodos Expandidos BFS : 682

Coste A\* : 54 Nodos Expandidos A\* : 535

Coste UCS : 54 Nodos Expandidos UCS : 682

DFS es el que tiene más coste y nodos expandidos, A\* el que tiene menos nodos expandidos y BFS y UCS tienen el mismo número de nodos expandidos. Estas tres últimas tienen el mismo coste.

**¿Qué funciona?**

Funciona DFS, BFS, UCS y A\*

**¿Qué no funciona?**

No funciona el problema de la búsqueda de puntos en las esquinas y por ende, su heurística.

**¿Qué problemas encontraron?**

La falta de conocimientos en Python no nos han permitido proseguir la práctica. Las “estructuras de datos” utilizadas en BFS no nos han permitido acabar de implementar un código funcional para la búsqueda de esquinas, por el hecho de que nosotros guardamos el nodo más las acciones para llegar a él en una tupla de la queue, pero para realizar el problema de las esquinas debíamos guardar más datos, como por ejemplo las esquinas que aún no habiamos encontrado. Esto se debería devolver en getStartState y en getSuccessors. Pero en getStartState de BFS solo se devuelve el nodo, y hemos sabido cómo acabar de añadir estos datos extras para que BFS siguiera funcionando correctamente fuera del problema de las esquinas.