-Práctica 1-
Inteligencia Artificial

Carlota Moncasi, 839841

Resultado obtenido tras ejecutar EightPuzzlePract.java:

Problema	Profundidad	Expand	Q.Size	MaxQS	tiempo
BFS-G-3	3	5	4	5	7
BFS-T-3	3	6	9	10	1
DFS-G-3	59123	120491	39830	42913	808
DFS-T-3	---	---	---	---	(1)
DLS-9-3	9	10	0	0	1
DLS-3-3	3	4	0	0	0
IDS-3	3	9	0	0	1
UCS-G-3	3	16	9	10	2
UCS-T-3	3	32	57	58	0
BFS-G-9	9	288	198	199	2
BFS-T-9	9	5821	11055	11056	15
DFS-G-9	44665	141452	32012	42967	643
DFS-T-9	---	---	---	---	(1)
DLS-9-9	9	5474	0	0	26
DLS-3-9	0	12	0	0	0
IDS-9	9	9063	0	0	13
UCS-G-9	9	385	235	239	4
UCS-T-9	9	18070	31593	31594	42
BFS-G-30	30	181058	365	24048	681
BFS-T-30	---	---	---	---	(1)
DFS-G-30	62856	80569	41533	41534	402
DFS-T-30	---	---	---	---	(1)
DLS-9-30	0	4681	0	0	7
DLS-3-30	0	9	0	0	0
IDS-30	---	---	---	---	(1)
UCS-G-30	30	181390	49	24209	669
UCS-T-30	---	---	---	---	

El algoritmo DFS-T o Deepening First Search en árbol conduce a un error en la ejecución, ya que necesita mucho espacio en memoria y por consiguiente, más tiempo del que se tiene. El mismo problema ocurre con el algoritmo BFS-T-30, en este caso debido a la profundidad máxima de 30, que es muy grande. Esto obliga a tener una larga cola de nodos explorados y, en consecuencia, un gran espacio de memoria, por lo que el programa se queda atascado. El IDS o Iterative Deepening Search también lleva a error, porque tiene una gran profundidad máxima, 30, que requiere mucha memoria.

BFS es ideal para encontrar la solución más corta, pero puede requerir mucha memoria, mientras que DFS puede encontrar soluciones más rápido pero no es necesariamente la más corta.