**Homework 4/4**

Question 4 of the general problem case description.

**Pregunta 4 de la descripción general del problema**

Utilizaremos la heurística del problema relajado donde tenemos en cuenta que todas las casillas son negras, y realizamos el máximo número de movimientos posibles. En este caso realizaremos cuatro acciones por ser todas negras. La heurística es admisible porque reducimos el número de acciones y el coste de la solución es menor que el coste real.

**Pregunta 5 de la descripción general del problema**

La búsqueda local analiza el estado actual y sus sucesores. No guarda nada en la lista Frontier para no retornar por caminos anteriores, esto quiere decir que no es completo. En cambio, el algoritmo de búsqueda clásica si guarda los nodos anteriores para poder retomarlos si no encuentra una buena solución, por tanto, es un algoritmo completo.

El algoritmo de búsqueda local genera un estado inicial de manera aleatoria, en cambio el algoritmo de búsqueda local no lo realiza. El algoritmo de búsqueda local puede que no encuentre ninguna solución y no devolverá ninguna solución, que será un problema. También tiene la opción de "reiniciar" que permitirá generar un nuevo estado inicial de manera aleatoria si se percibe que no se va a encontrar una buena solución.

**Punto 3**

Realizamos una comparación entre Best First Search, Depth First Search y Breadth First Search en base al tiempo, a las operaciones que realiza y al número de pasos según el orden de los operadores.

En base al tiempo:

|  |  |  |  |
| --- | --- | --- | --- |
| Operandos | Breadth First Search | Depth First Search | Best First Search |
| 2, 4, 1 | 2618000ns | 761000ns | 1019000ns |
| 1, 2, 4 | 2694000ns | 805000ns | 1168000ns |
| 4, 1, 2 | 2694000ns | 843000ns | 1181000sns |

En base a las operaciones que realiza:

|  |  |  |  |
| --- | --- | --- | --- |
| Operandos | Breadth First Search | Depth First Search | Best First Search |
| 2, 4, 1 | 144144 | 14124141 | 2222424 |
| 1, 2, 4 | 144144 | 2222424 | 1442241 |
| 4, 1, 2 | 144144 | 222212214 | 1442241 |

En base al número de pasos:

|  |  |  |  |
| --- | --- | --- | --- |
| Operandos | Breadth First Search | Depth First Search | Best First Search |
| 2, 4, 1 | 6 | 8 | 7 |
| 1, 2, 4 | 6 | 7 | 7 |
| 4, 1, 2 | 6 | 9 | 7 |

En el algoritmo de BreadthFS realiza las mismas operaciones con el mismo tiempo y mismos pasos.

En el algoritmo DepthFS se realizan diferente número de pasos dependiendo del orden de los operandos y el orden de las operaciones es diferente.

El algoritmo BestFS realiza el mismo número de acciones

**Punto 5**

Realizamos la comparación con el algoritmo de búsqueda A\* y los algoritmos de búsqueda Best First Search, Depth First Search y Breadth First Search, que hemos analizado en el punto anterior.

El algoritmo de búsqueda A\* tiene las siguientes características según el orden de los operandos.

|  |  |  |  |
| --- | --- | --- | --- |
| Operandos | Tiempo | Operaciones | Pasos |
| 2, 4, 1 | 3948000ns | 144144 | 6 |
| 1, 2, 4 | 3755000ns | 144144 | 6 |
| 4, 1, 2 | 3962000ns | 144144 | 6 |

El algoritmo A\* tarda mucho más tiempo que el resto de algoritmos, realiza siempre las mismas operaciones como el Breadth First Search y también realiza el mismo número de pasos que este último.

**Punto 6**

El algoritmo de búsqueda A\* no elimina estados repetidos, los deja al final de la lista Frontier y los va reordenando. Por ello, el tiempo que tarda el algoritmo Best First Search es bastante menor que el que tarda el algoritmo A\*. El algoritmo A\* tiene que cumplir la heurística de admisibilidad (es completo y óptimo) porque sino no encuentra una solución, es decir, la estimación tiene que ser mejor que el valor real. En conclusión, el algoritmo A\* es parecido al algoritmo Best First Search pero evita expandir todos los nodos.