**Endika Salgueiro Barquin**

**Javier Prada Leizea**

**SI-02**

**Sistemas Inteligentes**

**WALKING OVER A LINE OF BLACK & WHITE SQUARES**

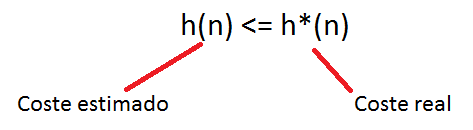
**Pregunta 4:**

La heurística que proponemos es la siguiente:

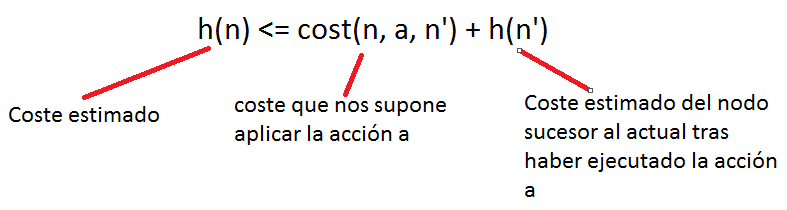
H(n) = número de casillas que nos quedan / operador máximo (4). (En caso de que la heurística nos diera un número decimal redondearíamos siempre hacia el siguiente número: 1.25🡪2; 5.86🡪6…).

Para definir esta heurística, hemos utilizado la técnica del problema relajado. Lo primero que hemos hecho ha sido imaginarnos que todo el tablero está formado por casillas negras. Dentro de cada casilla, hemos cogido el operador más grande, 4. Es decir, tenemos un tablero con todas las casillas negras en las que sólo podemos movernos de cuatro en cuatro.

Para saber si nuestra heurística es o no admisible, hemos comprobado que el coste estimado es menor o igual al coste real. Como se cumple esta regla, podemos afirmar que nuestra heurística es admisible.



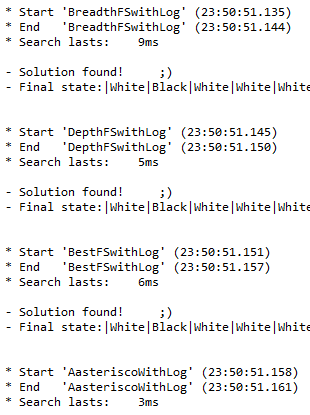
Para saber si nuestra heurística es o no Consistente, hemos comprobado que el coste estimado sea menor a la suma del coste que nos supone el operador que hemos ejecutado y el coste estimado del nodo que hemos obtenido al ejecutar el operador.



**Pregunta 5:**

De los algoritmos que hemos visto y comparado, nos quedaríamos con el algoritmo A\*.

El algoritmo A\* nos encuentra la solución en el menor número de movimientos posibles, es el más rápido y no necesitamos hacer la comprobación de estados repetidos. Es por todo ello por lo que lo hemos elegido.



**Otras preguntas:**

Si implementásemos el algoritmo de mejor primero sin comprobar estados repetidos conseguiríamos una versión simple de este algoritmo. Esta versión, seguiría sin poder compararse con la de A\* porque el algoritmo de mejor primero tiene que hacer la comprobación de estados repetidos entre otras cosas porque no tiene redireccionamiento.

En pocas palabras, para obtener el mejor rendimiento con el algoritmo de mejor primero deberíamos comprobar los estados repetidos. Sin embargo, con el algoritmo A\* no tenemos que hacer esta comprobación porque no tenemos esa necesidad al haber redireccionamientos.