

Perfecto 📖 Aquí tienes el texto original con **solo las faltas de ortografía corregidas**, sin cambiar estructura ni contenido:

1. La estrategia de búsqueda implementada con la función heurística Secuencias, ¿es un algoritmo A*? Justifica la respuesta.

```
python .\experiment.py .\data\10_puzzles.txt a_star_manhattan -> Solution Cost
20.80
python .\experiment.py .\data\10_puzzles.txt secuencia -----> Solution Cost
25.40
```

Al ser mayor secuencias miramos las posibles soluciones en el CSV y comparamos los pasos que dice la heurística (x) y los pasos reales hasta una solución P(x). Si $H(x) > P(x)$ en al menos un caso entonces sabemos que NO es admisible. Se ha encontrado en el que $H(x) > P(x)$ por lo tanto la heurística NO es admisible.

2. La estrategia de búsqueda implementada con la función heurística Filas Columnas, ¿es un algoritmo A*? Justifica la respuesta.

```
python .\experiment.py .\data\10_puzzles.txt a_star_manhattan ----> Solution Cost
20.80
python .\experiment.py .\data\10_puzzles.txt piezas_descolocadas -> Solution Cost
20.80
```

Podemos ver que ambas soluciones son iguales por lo que piezas_descolocadas no parece sobreestimar los pasos. --> Aquí Vicente mira el CSV generado por

```
python .\experiment.py .\data\10_puzzles.txt piezas_descolocadas
```

No he encontrado ningún caso en el que se sobreestime los pasos restantes con los predichos por la heurística. Por lo tanto, podemos asumir que esta heurística es admisible.

3. Compara la estrategia A* Manhattan con Secuencias e indica cuál de las dos estrategias devuelve mejores

soluciones (calidad de la solución y coste de la búsqueda).

Para que esta devuelva mejores soluciones ha de estar más informada (predice mayor número de pasos sin sobreestimarlos, para heurísticas \leq heurística). Por lo tanto, comparamos Solution cost. Este es igual, ya lo hemos visto en la pregunta anterior. Comparamos la cantidad de nodos generados ya que si está más informada necesitará generar menos nodos respecto a una menos informada. En este caso tenemos:

```
manhattan : Nodes Generated 905.60 (mean)
piezas_descolocadas : Nodes Generated 13387.0 (mean)
```

Aquí podemos ver una clara disonancia en la cantidad de nodos necesarios para encontrar una respuesta. Por lo cual podemos asumir que A* Manhattan está más informada que piezas_descolocadas.

4. Compara las estrategias de búsqueda implementadas con las heurísticas Piezas Descolocadas y Filas Columnas e indica cuál de las dos estrategias devuelve mejores soluciones (calidad de la solución y coste de la búsqueda)

Filas Columnas devuelve mejores soluciones:

- **Calidad de la solución:** Ambas son admisibles, por lo que las dos encontrarán soluciones óptimas.
- **Coste de búsqueda:** Filas Columnas es más informativa ya que considera la distancia en términos de filas y columnas incorrectas, no solo si la pieza está mal colocada. Esto proporciona una mejor estimación del coste real y resulta en una búsqueda más dirigida y eficiente.

Ambas son admisibles por lo cual devolverán soluciones óptimas. El método Filas y Columnas es mejor ya que este genera menos nodos y requiere menos tiempo total para resolverlo como se puede apreciar en estas tablas:

=====					
PERFORMANCE EVALUATION RESULTS					
=====					
Algorithm: filas_y_columnas					
PERFORMANCE METRICS	MEAN	MEDIAN	MIN	MAX	STD
DEV					

Nodes Generated	3639.50	3887	775	8065	

2635.18					
Nodes Expanded	1326.60	1411	281	2941	
961.88					
Max Nodes Stored	2157.20	2310	470	4732	
1543.80					
Solution Cost	20.80	22	16	24	2.35
Max Depth	20.80	22	16	24	2.35
Execution Time (s)	0.0183	0.0198	0.0050	0.0396	
0.0126					
Algorithm: piezas_descolocadas					
Total puzzles: 10					
PERFORMANCE METRICS	MEAN	MEDIAN	MIN	MAX	STD
DEV					

Nodes Generated	13387.00	15314	1975	33868	
9857.82					
Nodes Expanded	4945.90	5646	722	12513	
3646.17					
Max Nodes Stored	7682.80	8860	1183	18975	
5518.33					
Solution Cost	20.80	22	16	24	2.35
Max Depth	20.80	22	16	24	2.35
Execution Time (s)	0.0522	0.0583	0.0075	0.1296	
0.0382					