

### **Comparación de heurísticas**

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	0.15625	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic
Tipo de búsqueda	search	search	search	search	search

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	0.171875	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic
Tipo de búsqueda	search	search	search	search	search

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	0.140625	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic
Tipo de búsqueda	lazysearch	lazysearch	lazysearch	lazysearch	lazysearch

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	0.125	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic
Tipo de búsqueda	lazysearch	lazysearch	lazysearch	lazysearch	lazysearch

- Comparación entre las heurísticas "wagdy\_heuristic" y "repeated\_color\_heuristic":

Para todos los mapas (map\_1, map\_2, map\_3, map\_4 y map\_5), ambas heurísticas presentan tiempos de ejecución similares, con valores alrededor de 0.1 a 0.2 unidades.

Ambas heurísticas producen tiempos de ejecución iguales a "2 minutos y TIMEOUT" en todos los mapas, excepto en el mapa 1 donde el tiempo de ejecución es ligeramente menor para ambas heurísticas (0.15625 para "wagdy\_heuristic" y 0.171875 para "repeated\_color\_heuristic").

En términos de tipo de búsqueda, se utiliza el mismo enfoque de búsqueda "search" para ambas heurísticas.

- Comparación entre los tipos de búsqueda "search" y "lazysearch":

Para todos los mapas, tanto en el caso de "wagdy\_heuristic" como en "repeated\_color\_heuristic", se utiliza el tipo de búsqueda "lazysearch".

En términos de tiempo de ejecución, ambos tipos de búsqueda muestran resultados similares con valores alrededor de 0.1 a 0.2 unidades para todos los mapas.

- En conclusión:

Las heurísticas "wagdy\_heuristic" y "repeated\_color\_heuristic" tienen un rendimiento similar en términos de tiempo de ejecución, con tiempos ligeramente más bajos para "wagdy\_heuristic" en el mapa 1.

**Implementación con Best First Search y comparación**

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic
Tipo de búsqueda	search	search	search	search	search

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic
Tipo de búsqueda	search	search	search	search	search

---

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic	wagdy_heuristic
Tipo de búsqueda	lazysearch	lazysearch	lazysearch	lazysearch	lazysearch

Número de mapa	map_1	map_2	map_3	map_4	map_5
Tiempo de ejecución	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT	2 minutos y TIMEOUT
Tipo de heurística	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic	repeated_color_heuristic
Tipo de búsqueda	lazysearch	lazysearch	lazysearch	lazysearch	lazysearch

Acá al implementar GBFS, mis algoritmos no logran dar una en 2 minutos, por lo que las tabulé como *TIMEOUT*. Ahora, basandome en la teoría, podría explicar cuales son las grandes diferencias que tiene el GBFS con lazysearch:

- Greedy Best First Search se basa solo en la heurística y selecciona el nodo sucesor con el valor heurístico más bajo.
- Lazy A\* permite la duplicación de estados en la lista abierta y evita verificar y actualizar costos en cada paso.
- Greedy Best First Search no considera los costos acumulados y puede caer en óptimos locales.
- Lazy A\* utiliza una función de evaluación que tiene en cuenta el costo acumulado y la heurística.
- Lazy A\* puede encontrar soluciones rápidamente pero no garantiza la óptima debido a la duplicación de estados en la lista abierta.

Ahora, en general, GBFS tiende a ser mejor algoritmo que lazysearch, ya que su tiempo de ejecución generalmente es menor. Esto se debe a que Greedy Best First Search solo se basa en la heurística para tomar decisiones y no considera el costo acumulado hasta el momento. Por lo tanto, puede tomar decisiones rápidas y explorar rápidamente hacia las áreas que parecen más prometedoras según la heurística.