

# Rompecabezas de números SIA-TP1 2022 1C

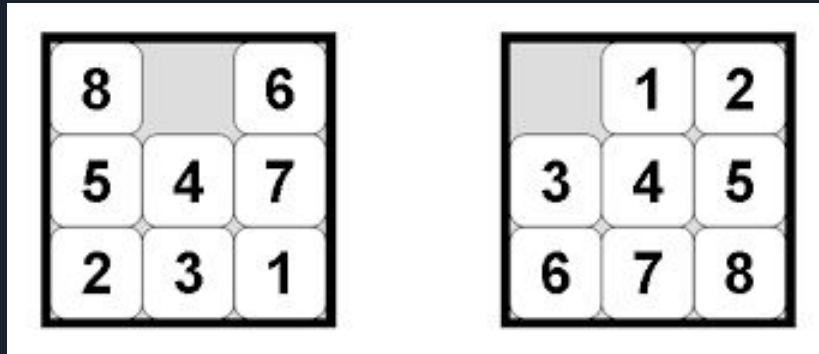
Sicardi, Julián Nicolás - Legajo 60347

Quintairos, Juan Ignacio - Legajo 59715

Zavalía Pángaro, Salustiano José - Legajo  
60312

# Descripción

- Tablero de dimensión  $N \times N$  con casilleros numerados de 1 a  $(N \times N) - 1$  y un espacio vacío.
- Objetivo: reordenarlas en orden ascendente.
- Movimientos: arriba, abajo, izquierda, derecha respetando bordes del tablero.



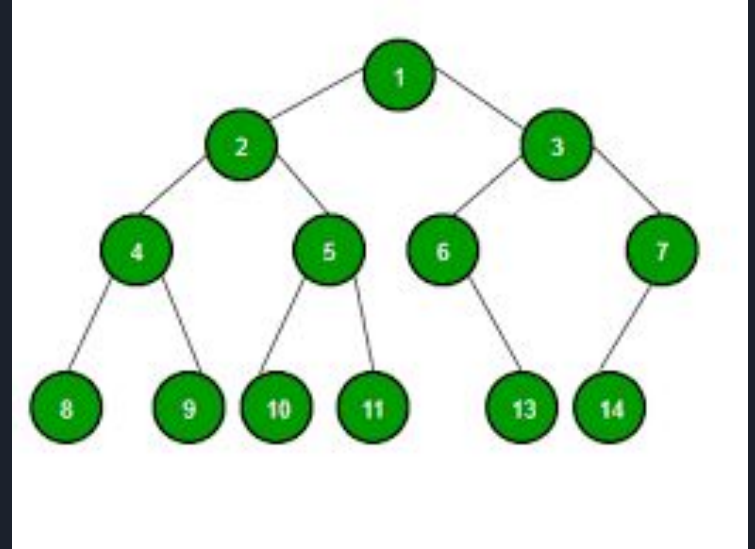
# Métodos de búsqueda

## Métodos no informados:

- BPP (DFS)
- BPA (BFS)
- BPPV (VDFS)

## Métodos informados:

- Heurística local con retroceso
- Heurística global
- A\*





# Configuración inicial

- Parámetros iniciales usando archivo config.json
- puzzle layout: tablero inicial
- algorithm: algoritmo a utilizar
- heurística: heurística a utilizar en métodos informados
- starting depth: profundidad inicial de método VDFS

```
{  
  "puzzle_layout": [[2,1,5],[3,6,4],[0,7,8]],  
  "algorithm": "VDFS",  
  "starting_depth": 20  
}
```

```
{  
  "puzzle_layout": [[2,1,5],[3,6,4],[0,7,8]],  
  "algorithm": "HEUR_LOCAL",  
  "heuristics": "total_manhattan"  
}
```



# Heurísticas

- **"total\_squares"**: Cantidad de números que no se encuentran en la posición correcta en el tablero. De esta forma se priorizan los estados con mayor cantidad de números en su lugar correcto.

Admisibles

- **"total\_manhattan"**: Distancia de manhattan para medir cuán lejos se encuentra un número de su posición correcta en el tablero. De esta forma se priorizan estados donde los números en conjunto estén más cerca de sus posiciones correctas.

No admisible

- **"total\_removing\_obstacles"**: Distancia de manhattan pero también considera que si hubieran números ocupando algunas de las casillas a recorrer se deberían gastar movimientos en correrlos para abrirle paso al número que estamos analizando

# Validación de estados iniciales

- Tablero de NxN
- Inversión: par de números en orden inverso al buscado en el resultado
- Es resoluble si
  - Para N impar, que la cantidad de inversiones sea par
  - Para N par, que la cantidad de inversiones sea par y el espacio vacío esté en fila impar (desde el fondo) o inverso

1	8	2
	4	3
7	6	5


8	1	2
	4	3
7	6	5

# Ejecución

- `python3 main.py`

```
Algorithm: HEUR_LOCAL
Heuristic: total_manhattan
Status: success
Depth: 396
Nodes expanded: 397
Nodes in frontier: 311
Time: 0.0293 s
Solution: see solution.txt
```

```
[1, 2, 3]
[4, 5, 6]
[7, 0, 8]
|
|
↓
[1, 2, 3]
[4, 5, 6]
[7, 8, 0]
```



## Análisis de resultados: Tableros para las muestras

**Tablero A:**

[ 2 1 5 ]

[ 3 6 4 ]

[ 0 7 8 ]

**Tablero B:**

[ 7 0 8 ]

[ 3 6 4 ]

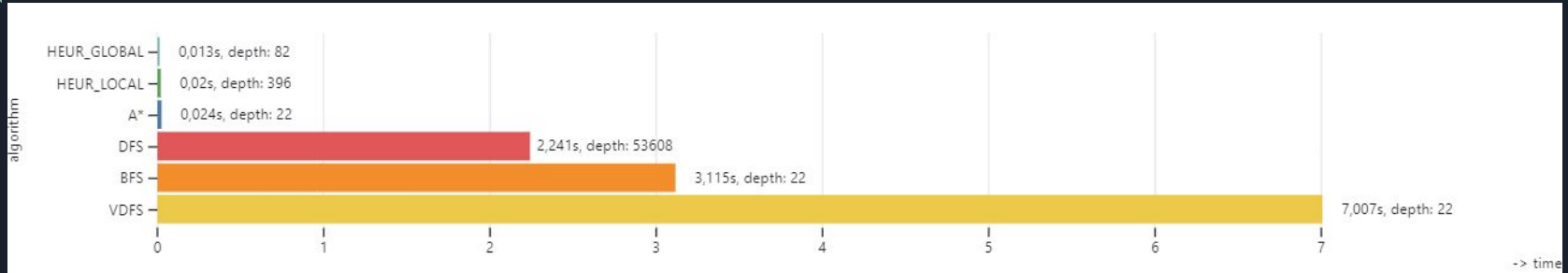
[ 2 5 1 ]

<https://observablehq.com/@a3152d54413ef2ed/tp-1-sia-2022q1>

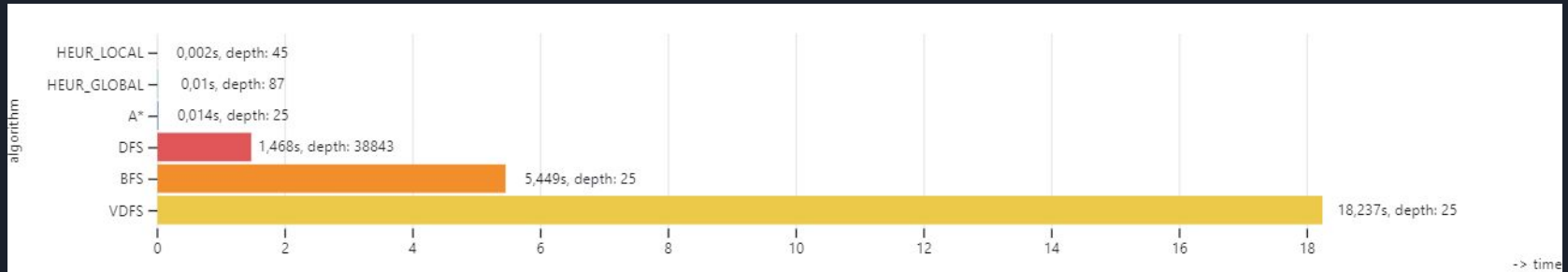


# Análisis de resultados: Tiempos de ejecución

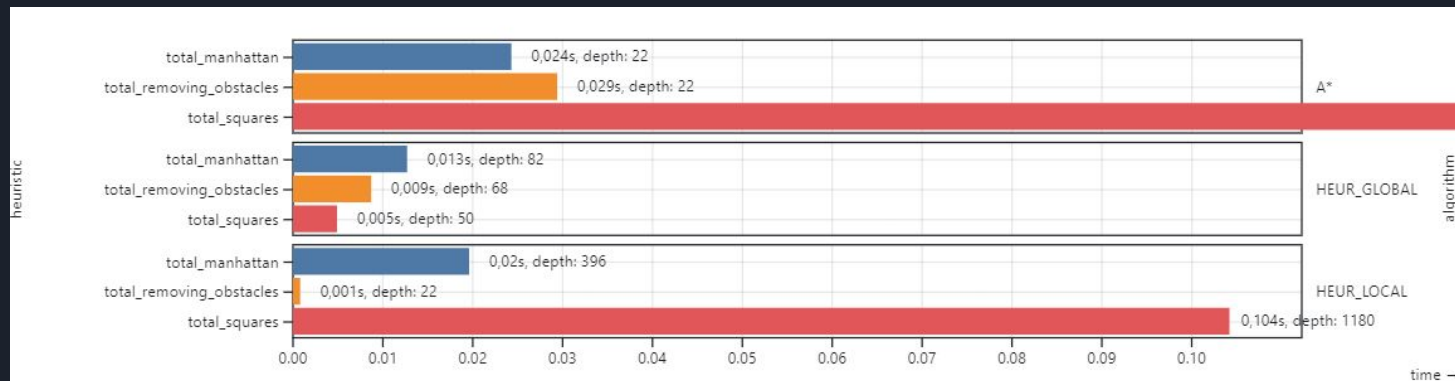
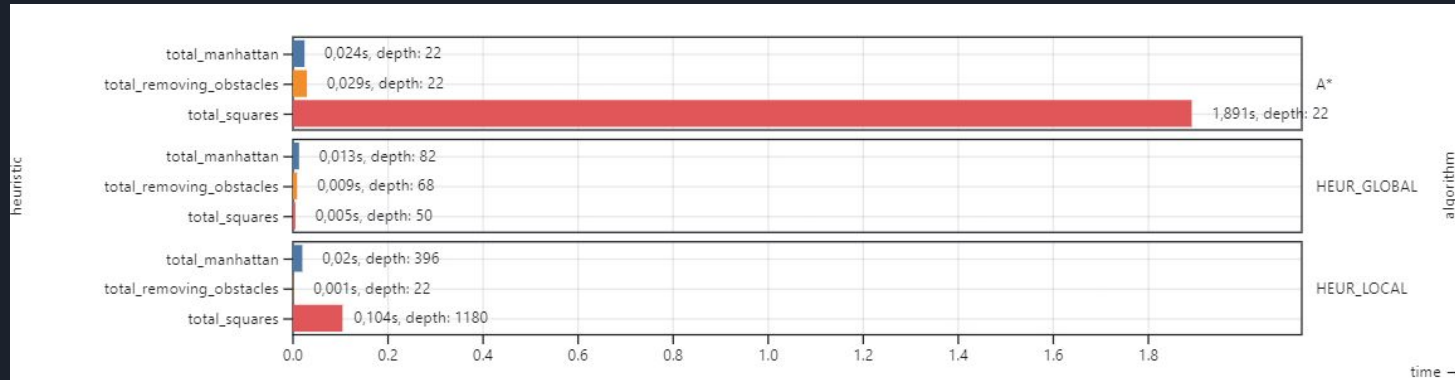
## Para tablero A



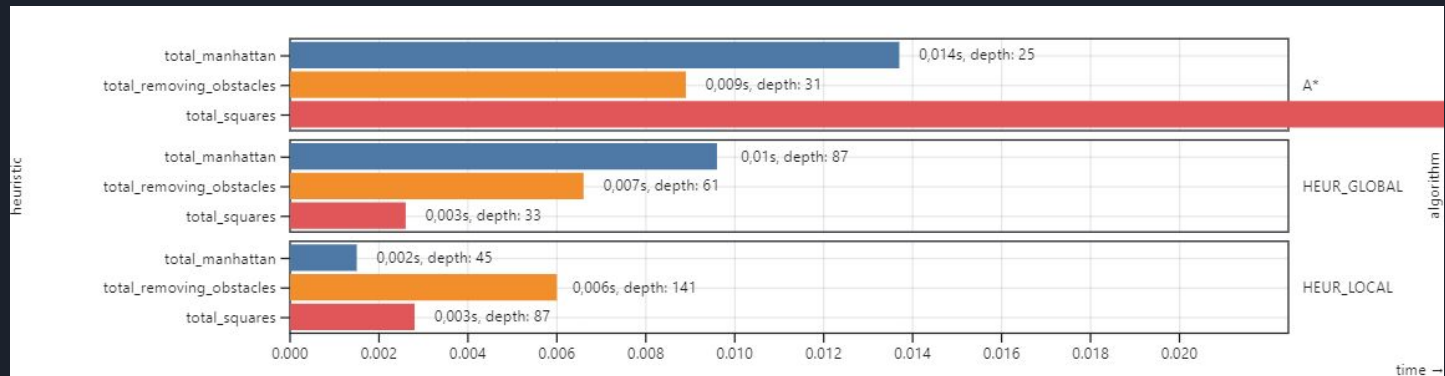
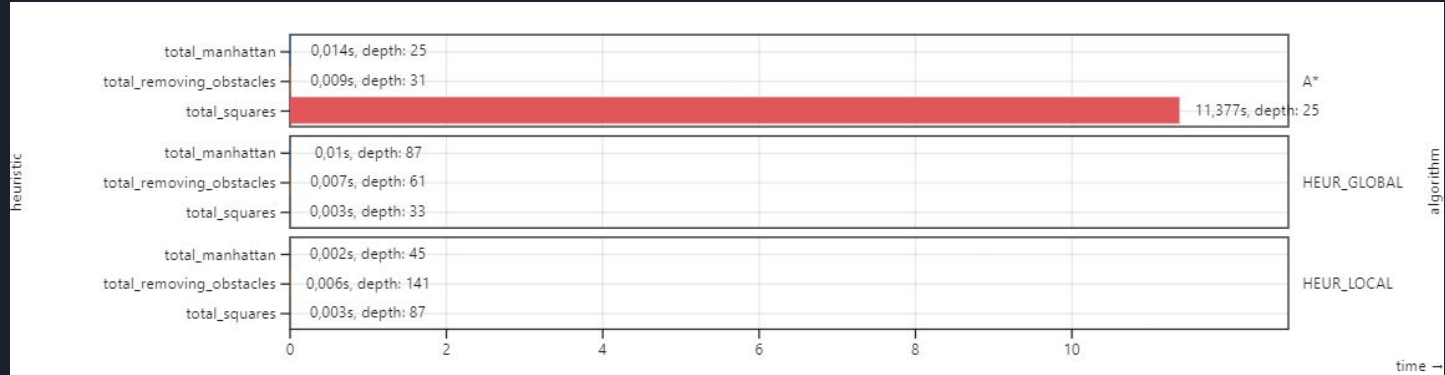
## Para tablero B



# Análisis de resultados: Tiempos de ejecución (Tablero A)

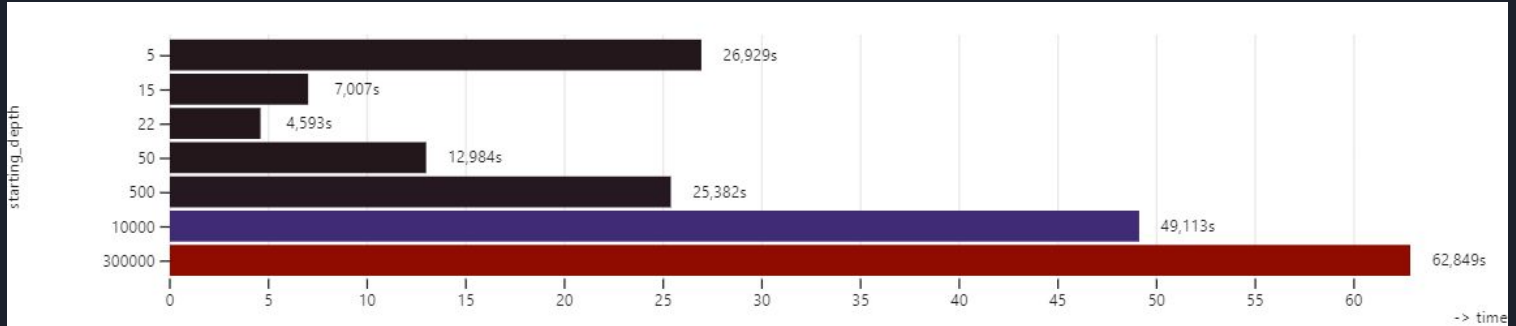


# Análisis de resultados: Tiempos de ejecución (Tablero B)

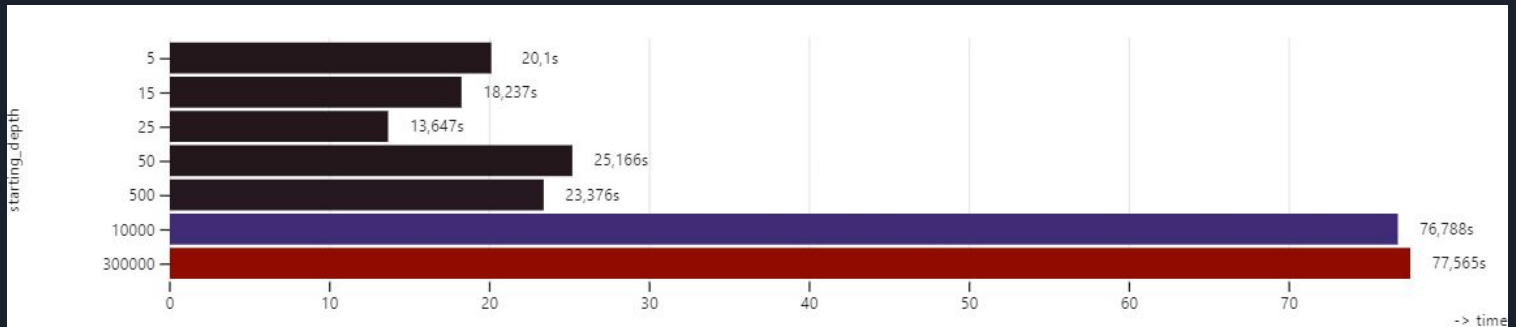


# Análisis de resultados: Tiempo de ejecución - VDFS

Para tablero A

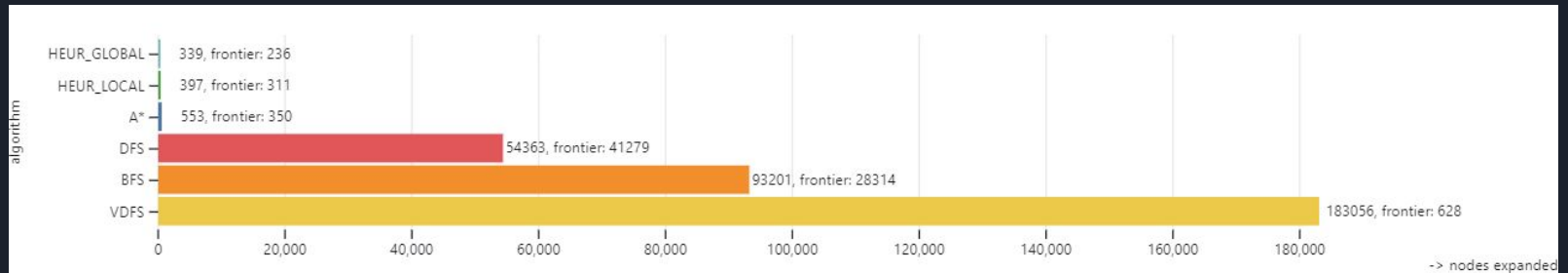


Para tablero B

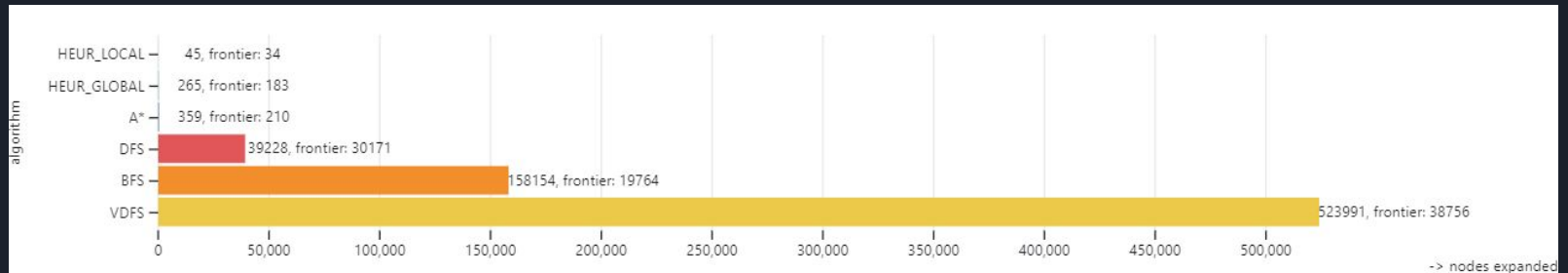


# Análisis de resultados: Nodos expandidos y longitud de frontera

Para tablero A



Para tablero B





¡Muchas  
Gracias!