

Trabajo Práctico 1:

Métodos de búsqueda no informados e informados

Grupo 4

Tomás Dallas

Tomás Dorado

Implementación

1. Sokoban
2. Estrategias no informadas
3. Estrategias informadas y heurísticas

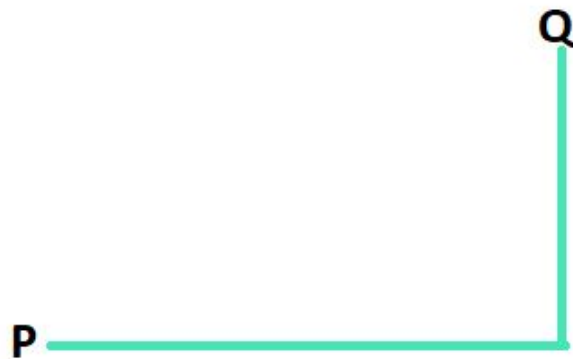
Heurísticas planteadas

- Mínima distancia local de Manhattan
- Mínima distancia local Euclidiana
- Distancia de Manhattan (muy) optimista
- Mínima distancia de Manhattan global
- Mínima distancia Euclidiana global

Distancia de Manhattan

Sea $P = (p_1, p_2)$ y $Q = (q_1, q_2)$ dos puntos cualesquiera:

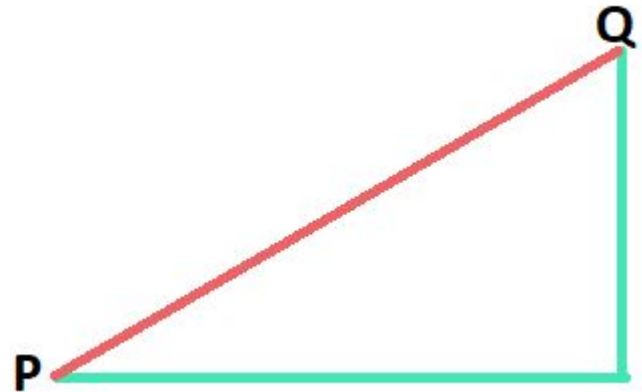
$$\text{Distancia de Manhattan} = |p_1 - q_1| + |p_2 - q_2|$$



Distancia Euclidiana

Sea $P = (p_1, p_2)$ y $Q = (q_1, q_2)$ dos puntos cualesquiera:

$$\text{Distancia Euclidiana} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$



Mínima distancia local

valorBase \leftarrow calcularDistanciaALaCajaMásCercana

sumatoriaDeDistanciaMinimaParaCajas \leftarrow 0

Para cada caja:

 sumatoriaDeDistanciaMinimaParaCajas $+\leftarrow$
 calcularDistanciaALaPosiciónFinalMásCercana

valorFinal \leftarrow valorBase + sumatoriaDeDistanciaMinimaParaCajas

Mínima distancia global

- Objetivo: Generar movimientos cortos
- En cada cálculo de mínima distancia se chequea contra un valor global si se encontró un mínimo. De ser así, ese mínimo se guarda y se usa posteriormente para los próximos cálculos de heurística.

Distancia (muy) optimista de Manhattan

- Se buscó ser optimista de más, calculando los valores mínimos para la mitad de las posibilidades.
- Si tengo 4 cajas, solo busco sobre la mitad de ellas la distancia mínima.
- La heurística confía en que la mitad de los cálculos darán una solución óptima.
- Equilibrio entre menos pasos y menos tiempo.

¿Es una heurística admisible?

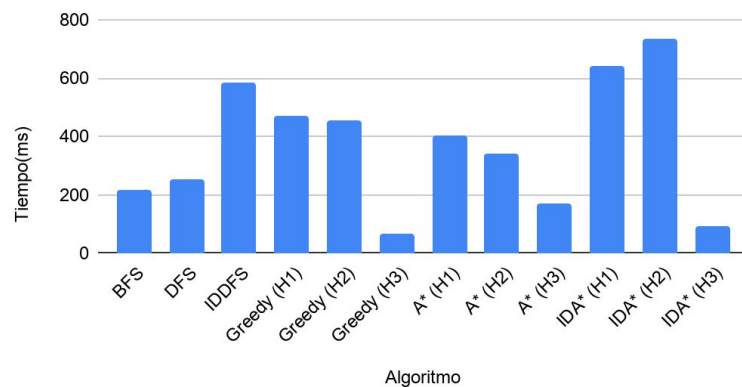
Tests

- Objetivo → 75%
- Líneas totales del proyecto ~ 1600 líneas

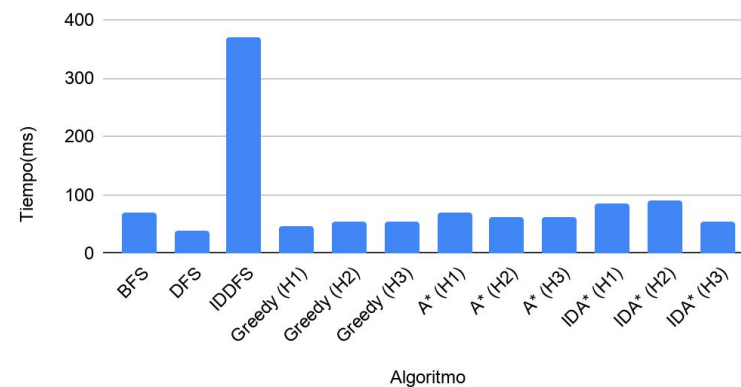
Element	Class, %	Method, %	Line, %
Class50			
com			
game	100% (13/13)	84% (42/50)	84% (169/200)
images			
java			
javax			
jdk			
lombok			
META-INF			
netbeans			
org			
strategies	100% (17/17)	74% (56/75)	85% (304/356)
sun			
toolbarButtonGraphics			
Main	0% (0/1)	0% (0/2)	0% (0/88)

Métricas

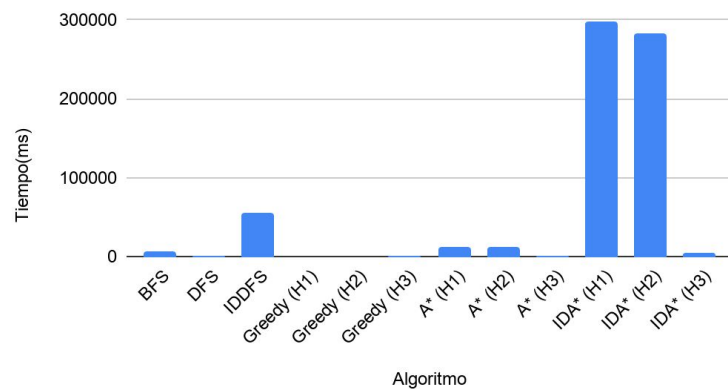
Tiempo(ms) frente a Algoritmo en Mapa 2



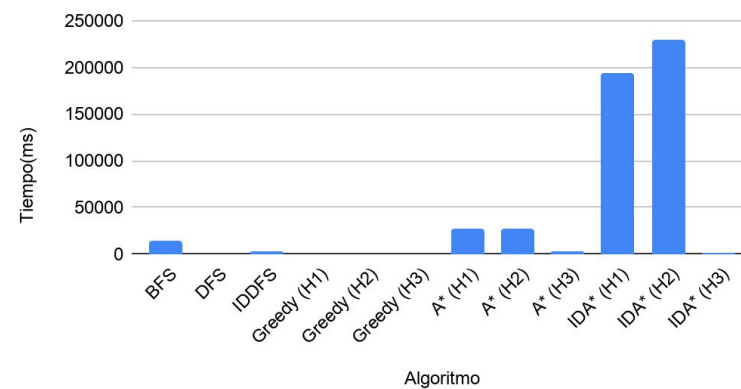
Tiempo(ms) frente a Algoritmo en Mapa 3



Tiempo(ms) frente a Algoritmo en Mapa 4



Tiempo(ms) frente a Algoritmo en Mapa 5



Métricas

MAPA 2 (maps/m5.txt)						
Algoritmo	Tiempo(ms)	Costo	Longitud	Nodos expandidos	Nodos frontera	Solución encontrada
BFS	216	23	23	5873	7364	SUCCESS
DFS	254	153	153	19152	9761	SUCCESS
IDDFS	586	31	31	99056	47798	SUCCESS
Greedy (H1)	470	499	499	12221	16926	SUCCESS
Greedy (H2)	454	499	499	12221	16926	SUCCESS
Greedy (H3)	69	25	25	886	1060	SUCCESS
A* (H1)	401	23	23	6541	8284	SUCCESS
A* (H2)	339	23	23	1570	2065	SUCCESS
A* (H3)	169	37	37	3936	4805	SUCCESS
IDA* (H1)	640	23	23	17226	26118	SUCCESS
IDA* (H2)	734	23	23	17226	26118	SUCCESS
IDA* (H3)	92	35	35	1926	2789	SUCCESS

MAPA 3 (maps/m6.txt)						
Algoritmo	Tiempo(ms)	Costo	Longitud	Nodos expandidos	Nodos frontera	Solución encontrada
BFS	69	69	69	1023	1132	SUCCESS
DFS	38	139	139	1128	553	SUCCESS
IDDFS	370	71	71	49483	26014	SUCCESS
Greedy (H1)	47	119	119	692	763	SUCCESS
Greedy (H2)	53	119	119	692	763	SUCCESS
Greedy (H3)	53	95	95	491	531	SUCCESS
A* (H1)	69	69	69	1032	1144	SUCCESS
A* (H2)	62	69	69	1032	1144	SUCCESS
A* (H3)	62	69	69	981	1078	SUCCESS
IDA* (H1)	85	69	69	1571	1881	SUCCESS
IDA* (H2)	91	69	69	1571	1881	SUCCESS
IDA* (H3)	54	83	83	833	978	SUCCESS

Métricas

MAPA 4 (maps/m7.txt)						
Algoritmo	Tiempo(ms)	Costo	Longitud	Nodos expandidos	Nodos frontera	Solución encontrada
BFS	6957	72	72	233621	325222	SUCCESS
DFS	254	262	262	29458	14586	SUCCESS
IDDFS	55538	194	194	14059724	7171460	SUCCESS
Greedy (H1)	101	228	228	2455	3303	SUCCESS
Greedy (H2)	100	228	228	2455	3303	SUCCESS
Greedy (H3)	987	212	212	31993	41756	SUCCESS
A* (H1)	11816	74	74	229990	320087	SUCCESS
A* (H2)	12009	74	74	229990	320087	SUCCESS
A* (H3)	1429	122	122	67292	88822	SUCCESS
IDA* (H1)	297103	74	74	6525631	11504379	SUCCESS
IDA* (H2)	282670	74	74	6525631	11504379	SUCCESS
IDA* (H3)	5776	150	150	240956	392990	SUCCESS

MAPA 5 (maps/m8.txt)						
Algoritmo	Tiempo(ms)	Costo	Longitud	Nodos expandidos	Nodos frontera	Solución encontrada
BFS	13974	26	26	270347	425489	SUCCESS
DFS	84	408	408	6902	3369	SUCCESS
IDDFS	2791	36	36	735613	288041	SUCCESS
Greedy (H1)	103	272	272	2296	3661	SUCCESS
Greedy (H2)	100	272	272	2296	3661	SUCCESS
Greedy (H3)	323	82	82	8718	12120	SUCCESS
A* (H1)	27224	30	30	279101	443737	SUCCESS
A* (H2)	27322	30	30	279101	443737	SUCCESS
A* (H3)	2507	52	52	80914	113318	SUCCESS
IDA* (H1)	194222	26	26	2503531	5046784	SUCCESS
IDA* (H2)	230177	26	26	2503531	5046784	SUCCESS
IDA* (H3)	1038	70	70	45066	70082	SUCCESS

Problemas encontrados

- Inconsistencia con el manejo de colecciones durante simulaciones de movimientos
- Al principio modelamos de una manera y no contemplamos la necesidad de simular movimientos rápida e intuitivamente
- Implementación de IDA*
- Pensar heurísticas

Conclusiones

Búsquedas no informadas:

- BFS fue sustancialmente el mejor en cuanto a tiempo y camino óptimo
- DFS fue muy rápido siempre, pero el camino no fue bueno nunca.
- IDDFS fue lento en varios mapas, pero con caminos no malos, pero tampoco óptimos. No lo vale comparado con BFS en general.

Conclusiones

Búsquedas informadas:

- Greedy siempre fue rápido, pero nunca devolvió el mejor camino.
- Las heurísticas **GlobalMinManhattan** y **GlobalMinEuclidean** fueron siempre las mejores usando A^* , comparadas con las demás, pero la ManhattanCheckinHalf en general superó a las otras, cuando se las utiliza con Greedy. Se podría decir que está en el medio de H1 y H2 con A^* y H1 y H2 con Greedy.
- Se puede ver como IDA^* con una heurística admisible siempre devuelve el mejor camino, pero en tiempo puede tardar mucho.

Conclusiones

- Escalabilidad y robustez del proyecto.
- Se podrían utilizar en el futuro técnicas de aprendizaje supervisado (redes neuronales) dado que el problema es NP-hard y “poco” eficiente para algoritmos de fuerza bruta.
- Mejorar heurísticas y mapas más complejos