
Ayudantía 2

Diego Caroca : dncaroca@uc.cl
Sergio M Appel : matamalaappels@uc.cl

A*

- Algoritmo de búsqueda informada
 - $f(s) = g(s) + h(s)$
 - $h(s)$ = función heurística.
- if $h(s)$ admisible y consistente \rightarrow A* siempre encuentra el óptimo

Heurística

Admisibilidad :

- $h(s) \leq h^*(s) \quad \forall \quad s$
 - Una heurística nunca sobrestima el costo para llegar al estado objetivo.

Consistencia :

- $h(s) = 0 \quad \forall \quad s$ subconjunto de G
 - La heurística vale 0 en el objetivo. (Goal)
- $h(s) \leq c(s, s') + h(s') \quad \forall \quad$ vecino s' de s
 - $s = \text{Padre}, s' = \text{Hijo}$

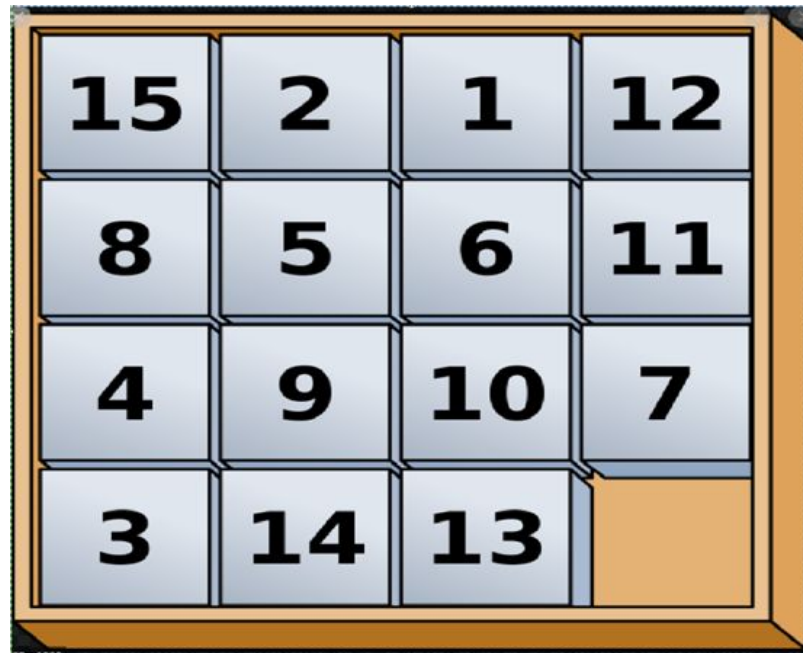
Consistencia + Admisibilidad -> Garantiza optimalidad en la solución

#prob	#exp	#gen	sol	tiempo	maxsubopt	#prob	#exp	#gen	sol	tiempo	maxsubopt
1	118	249	1.51	0.05	1.00	1	131	280	1.51	0.05	1.00
2	1457	2976	2.05	0.20	1.00	2	1843	3764	2.05	0.16	1.00
3	2859	5956	2.18	0.39	1.00	3	2987	6421	2.18	0.20	1.00
4	10623	20473	2.80	1.16	1.00	4	9725	18531	2.80	0.64	1.00
5	3355	6871	2.59	0.42	1.00	5	6575	13456	2.59	0.42	1.00
6	5383	10978	2.66	0.64	1.00	6	5164	10606	2.66	0.39	1.00
7	19047	37255	3.17	2.12	1.00	7	20200	39803	3.17	1.36	1.00
8	8429	17118	2.67	1.02	1.00	8	12953	26121	2.67	0.81	1.00
9	12575	25115	2.63	1.45	1.00	9	10060	19950	2.63	0.66	1.00
10	28279	53053	3.62	3.09	1.00	10	27608	50788	3.62	1.69	1.00

Relajación

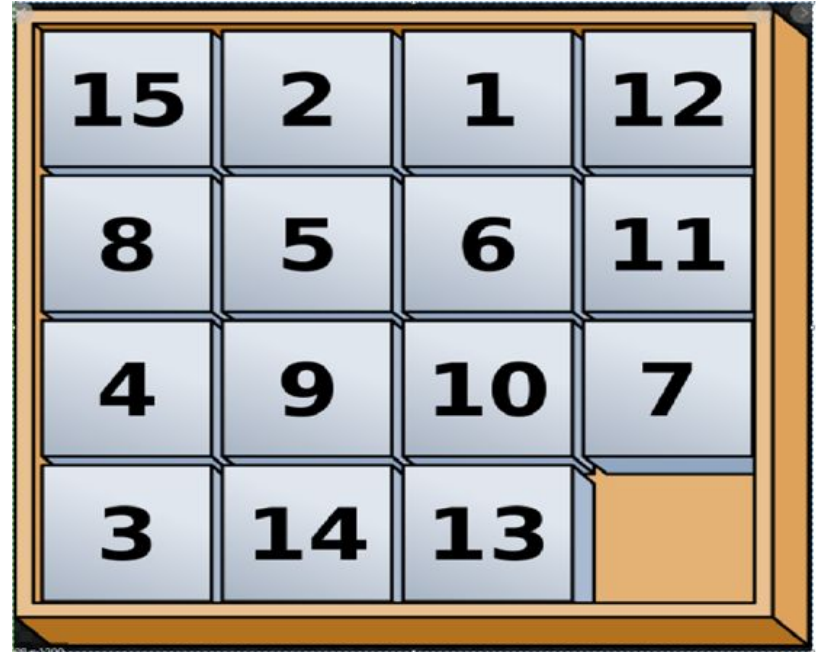
- Metodología para abordar los problemas
 - Queremos “encontrar” una mejor heurística
- Tomamos un subconjunto de G
- Obviamos restricción/es.
- Partimos con la base de una heurística \rightarrow una heurística más informada
 - $h_i(s)$ if $\forall s [h_i(s) \leq h_j(s)]$ and $\exists s [h_i(s) < h_j(s)]$ * ssi ambas son admisibles
 - Intentamos abordar ese problema \rightarrow llegar a una heurística más informada

Puzzle de 15

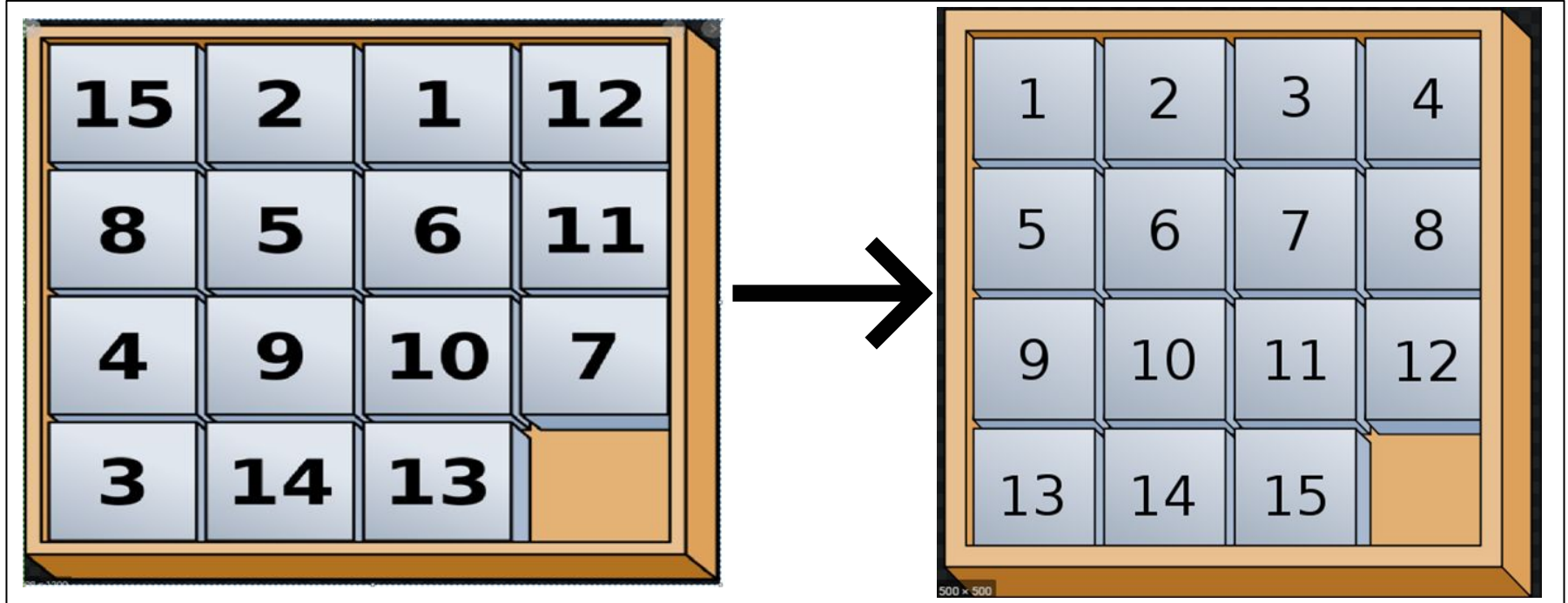


Cómo lo resolvemos ?

- Manhattan
- ??
- ??



Práctico



5		3

	1	2
3	4	5
6	7	8

Estado objetivo

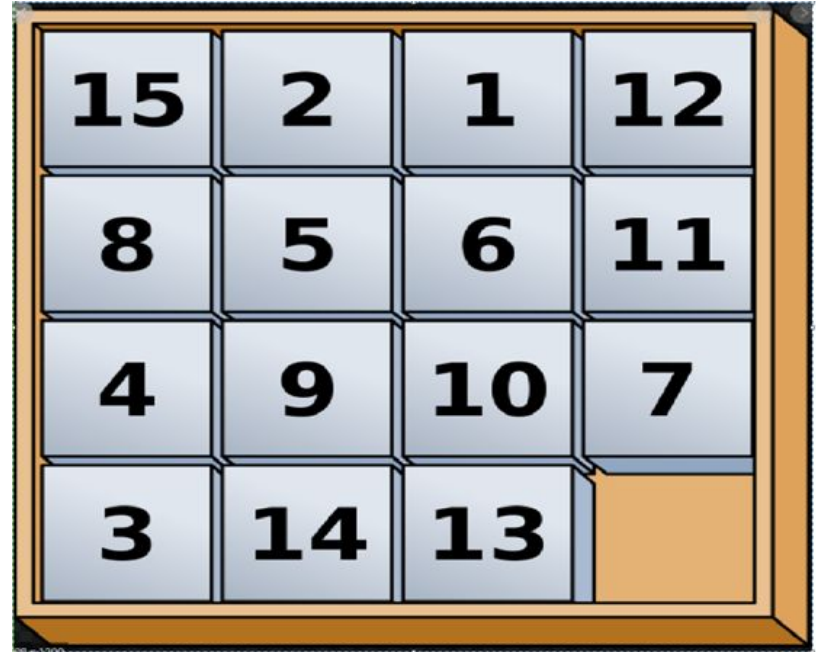
5	4	3

	1	2
3	4	5
6	7	8

Estado objetivo

Cómo lo resolvemos ?

- Manhattan
- Conflictos Lineales



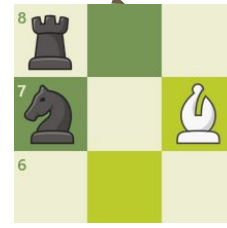
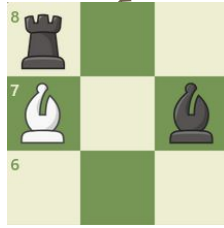
Búsqueda con adversario

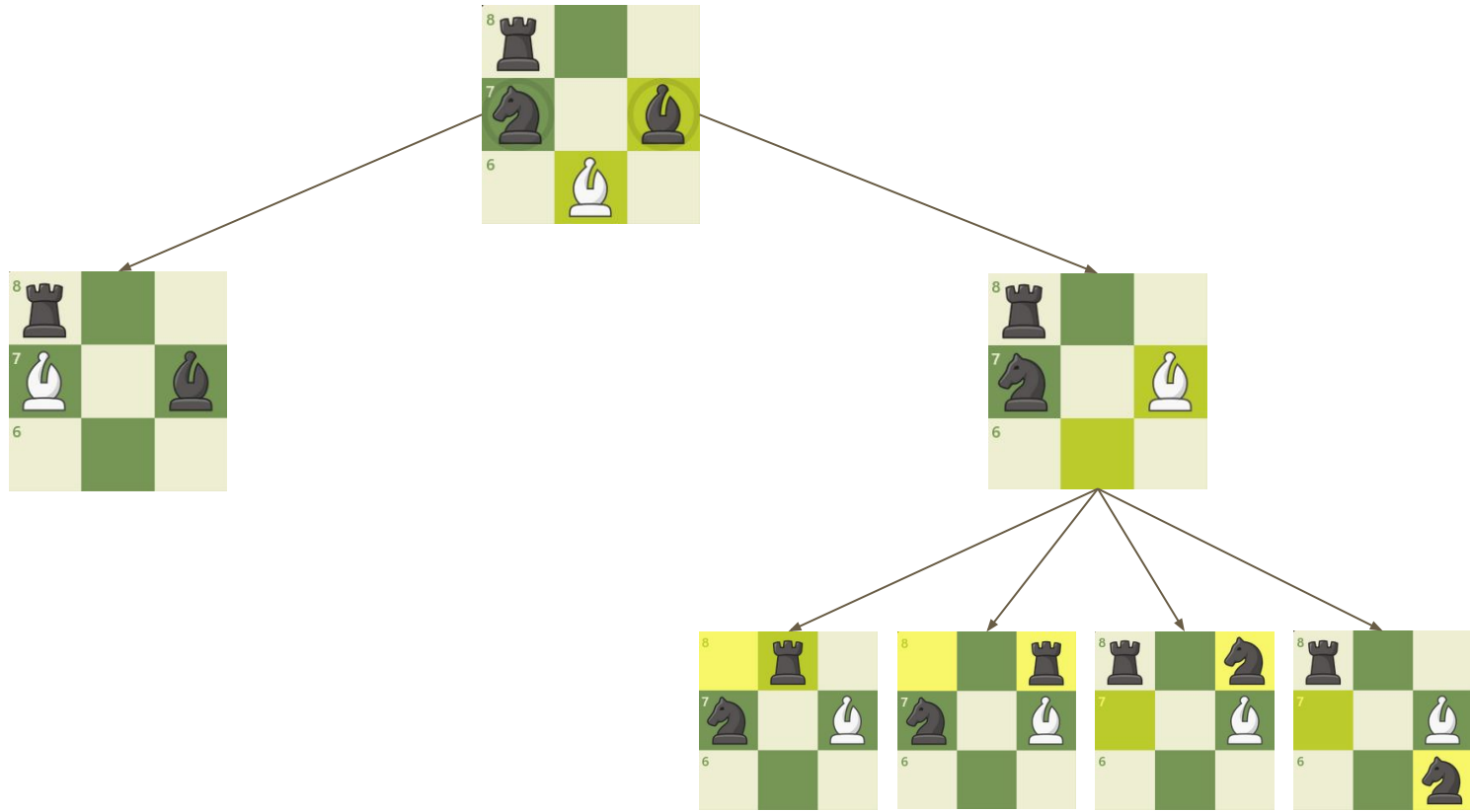
- Árboles minimax para alternar entre la mejor acción para el jugador (maximizar) y la mejor para el oponente (minimizar)

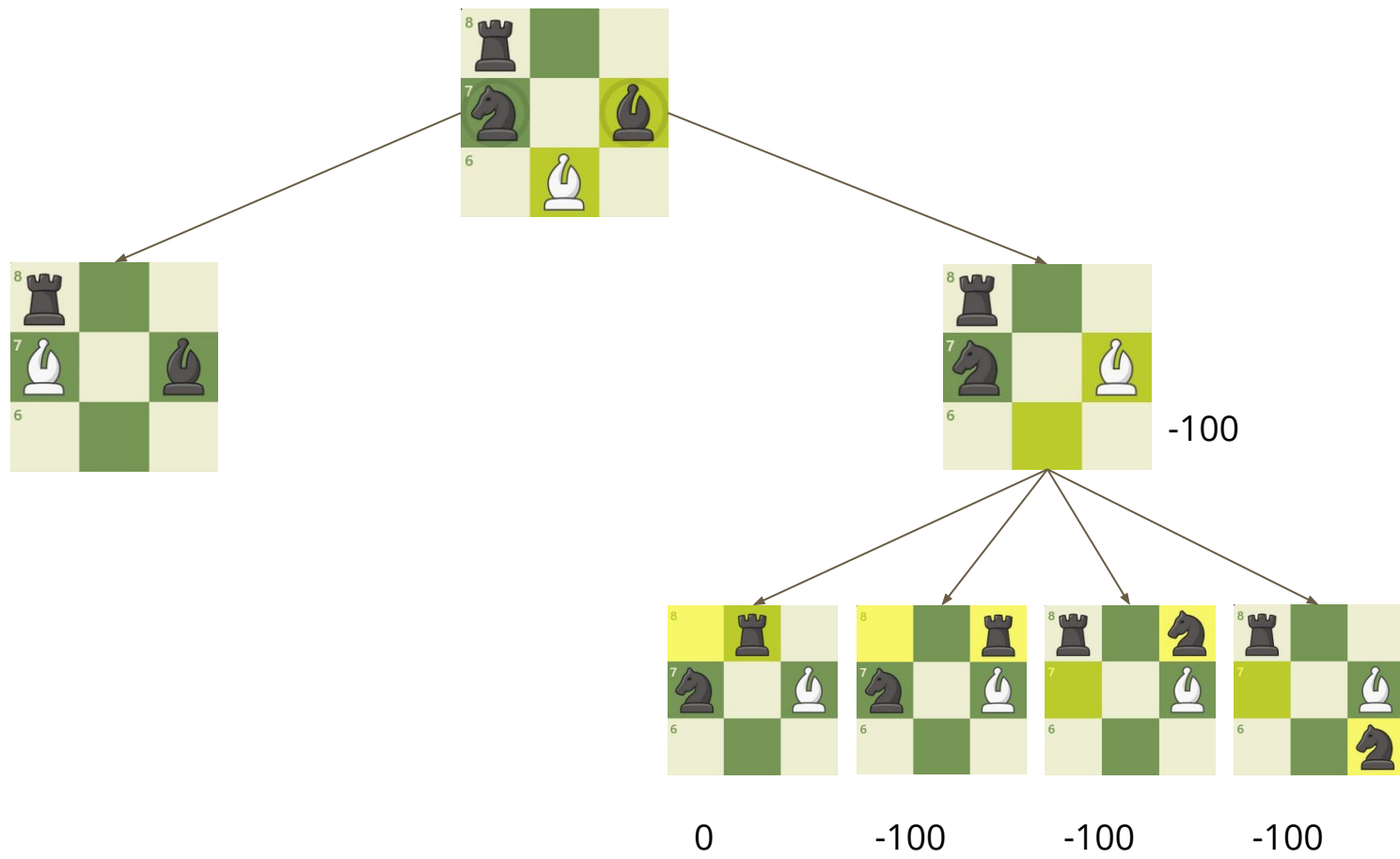
Ejemplo - Ajedrez

- Tablero reducido (3x3)
- Juegan las blancas
- El alfil tiene sólo 2 movimientos posibles: comer al caballo y comer al alfil



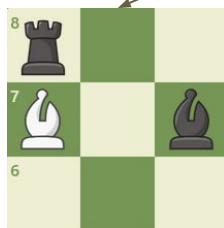




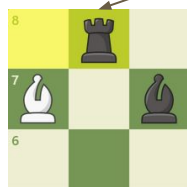




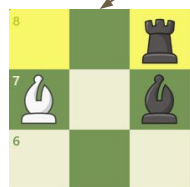
-100



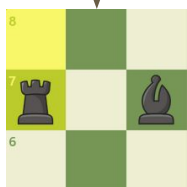
-200



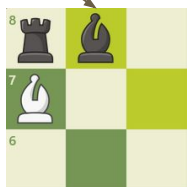
-100



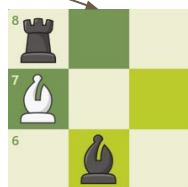
-100



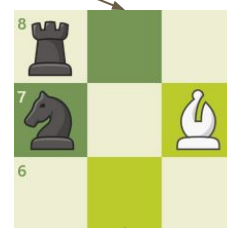
-200



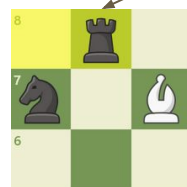
-100



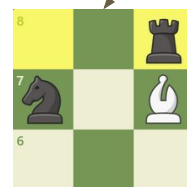
-50



-100



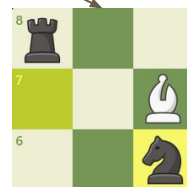
0



-100



-100



-100

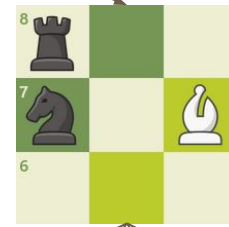
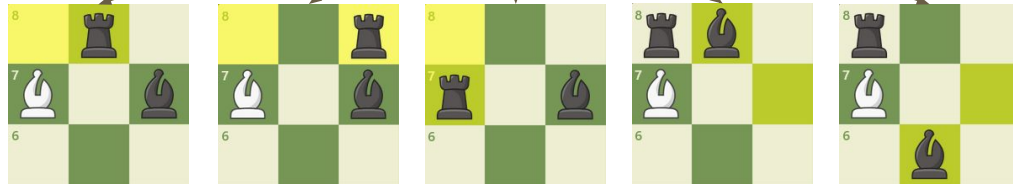
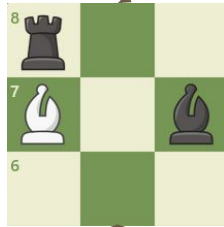
Poda alfa-beta

- Quitar una rama cuando he encontrado un mejor resultado a través de otra
- Se realiza comparando el mejor hasta ahora y el peor hasta ahora desde dos niveles distintos (pero adyacentes) del árbol.

Mismo ejemplo de antes - Ajedrez

- Tablero reducido (3x3)
- Juegan las blancas
- El alfil tiene sólo 2 movimientos posibles: comer al caballo y comer al alfil





-100

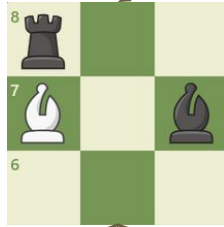


0

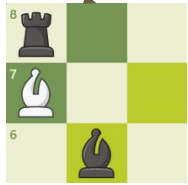
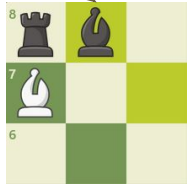
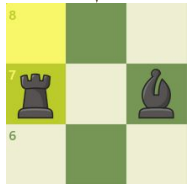
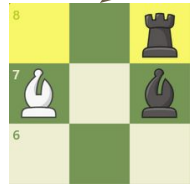
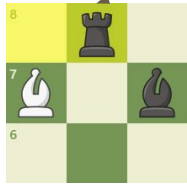
-100

-100

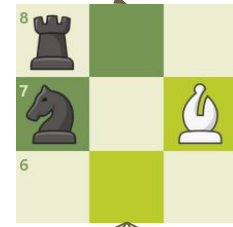
-100



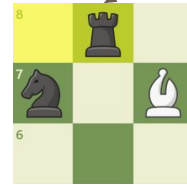
a lo más, -100



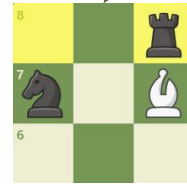
-100



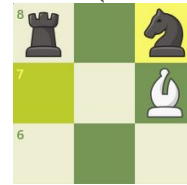
-100



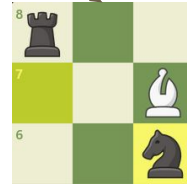
0



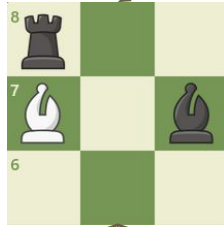
-100



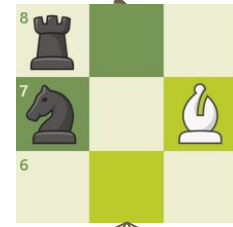
-100



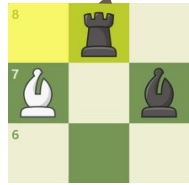
-100



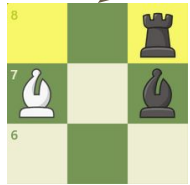
a lo más, -100



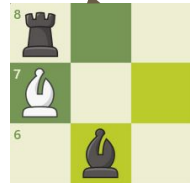
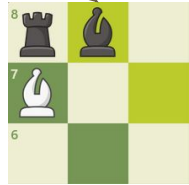
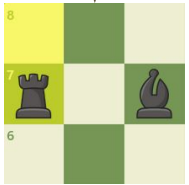
-100



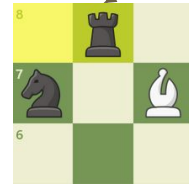
-100



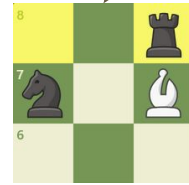
-100



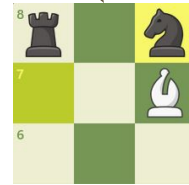
.....



0



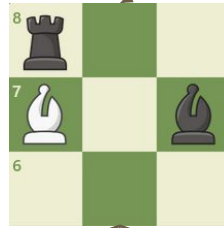
-100



-100



-100



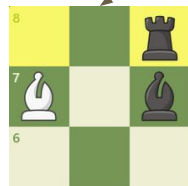
a lo más, -200



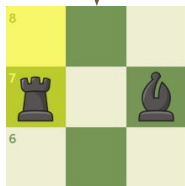
-100



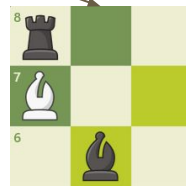
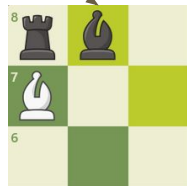
-100



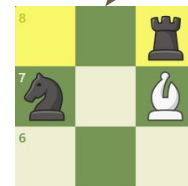
-100



-200



0



-100



-100



-100

