

# Tarea 8 Puzzle

Saul Ivan Rivas Vega

Inteligencia Artificial

9 de septiembre de 2019

En el presente trabajo se describen tres búsquedas a partir de un estado inicial a un estado final en el puzzle 8, un rompecabezas deslizante que tiene piezas numeradas del 1 al 8 en un tablero de 3 x 3, lo cual hace que tenga un hueco y sea precisamente este el que permita que las piezas se deslicen y ocupen el lugar del hueco.

El *estado inicial* para el ejercicio es el siguiente:

4	2	1
7	3	5
	8	6

Y el *estado final*:

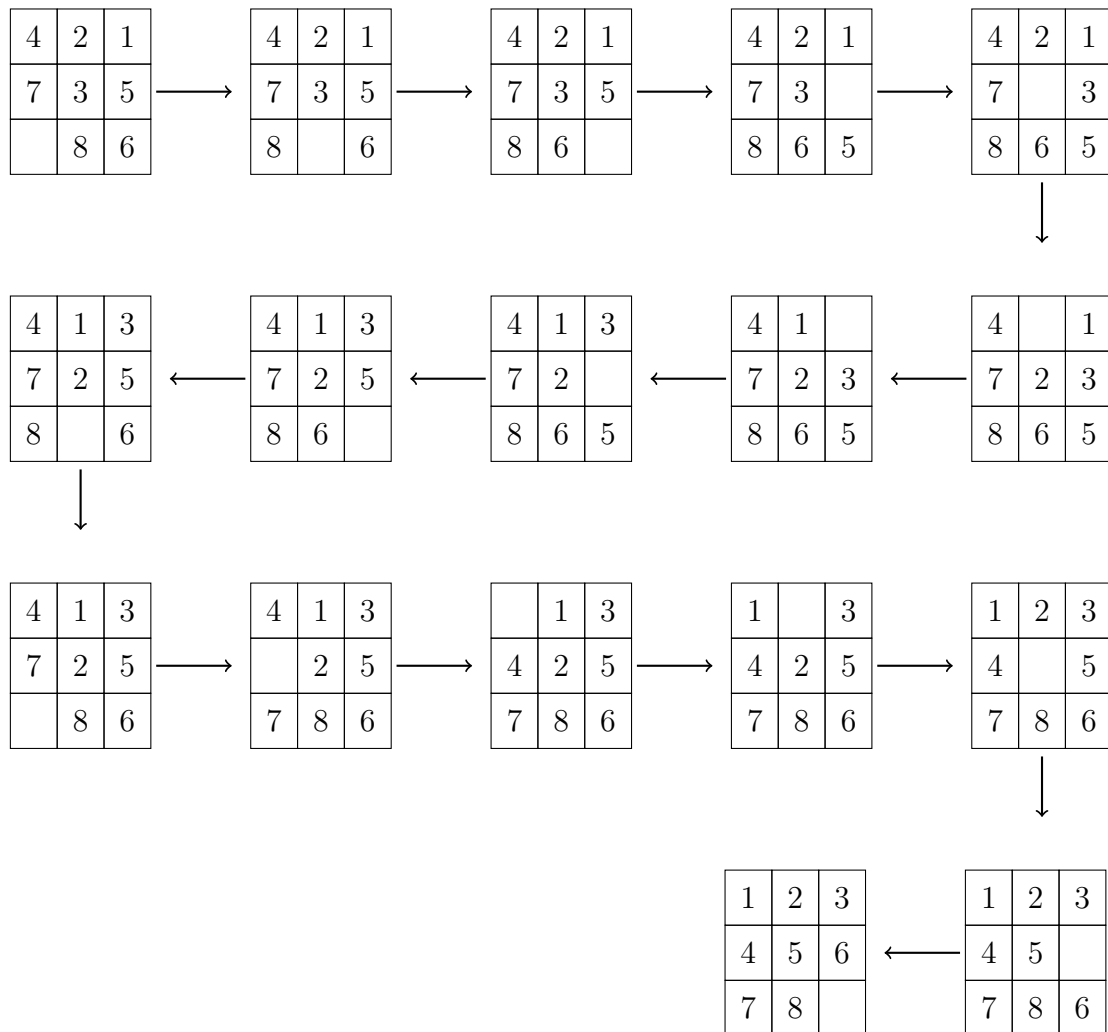
1	2	3
4	5	6
7	8	

Las piezas que se pueden mover son las que son adyacentes al hueco. Un estado del rompecabezas  $a$  es *vecino* de otro estado  $b$  si de  $a$  se puede llegar a  $b$  en un movimiento.

### Métodos de búsqueda:

- Búsqueda en amplitud.
- Búsqueda en profundidad.
- Búsqueda por Ascensión de colinas (Hill Climbing).

**Camino de solución.** Este es una sucesión de estados vecinos que llevan del *estado inicial* al *estado final*.



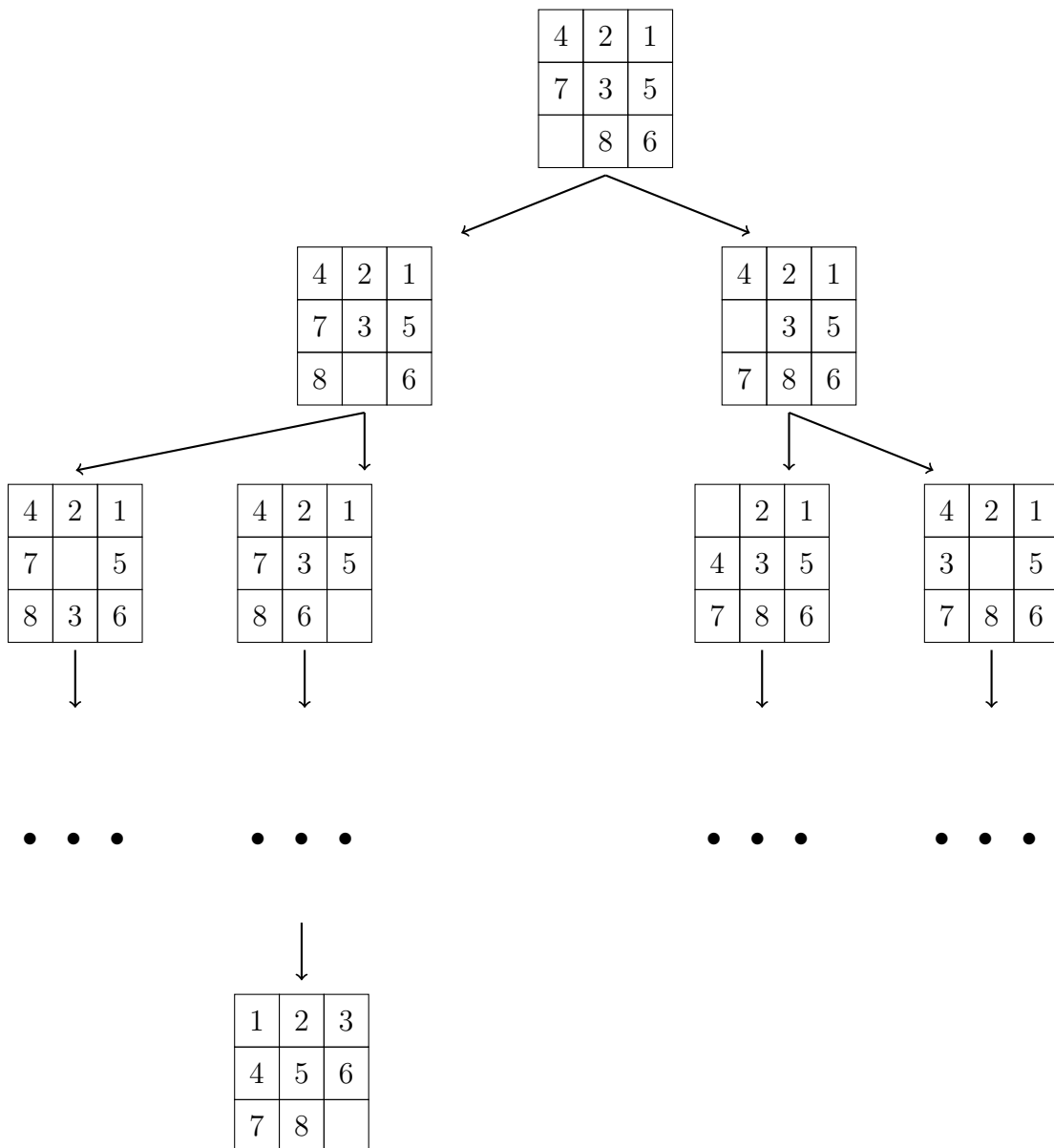
## 1. Búsqueda en Amplitud.

En este método se toma un *estado inicial*, se inserta en una estructura de datos que cumpla la propiedad de que el primero que entra es el primero que sale, es decir una *cola*  $q$ . Ahora mientras que haya elementos en  $q$  se realizarán las siguientes operaciones.

- Sacar  $u$ , siendo el primer elemento formado en  $q$ .
- Si  $u$  es igual al *estado final*, terminar.
- Revisar cuales de sus vecinos  $v$  no han sido visitados e insertarlos en  $q$ .

El método anterior recorrerá los estados en orden por la propiedad de orden de  $q$ , lo que es recorrer los nodos en orden de distancia al *estado inicial*. Se recorrerán los estados a distancia 1, después los de distancia 2 así sucesivamente hasta llegar al *estado final*. Visto en una gráfica con estados como vértices:

Gráfica búsqueda en amplitud.



## 2. Búsqueda en Profundidad.

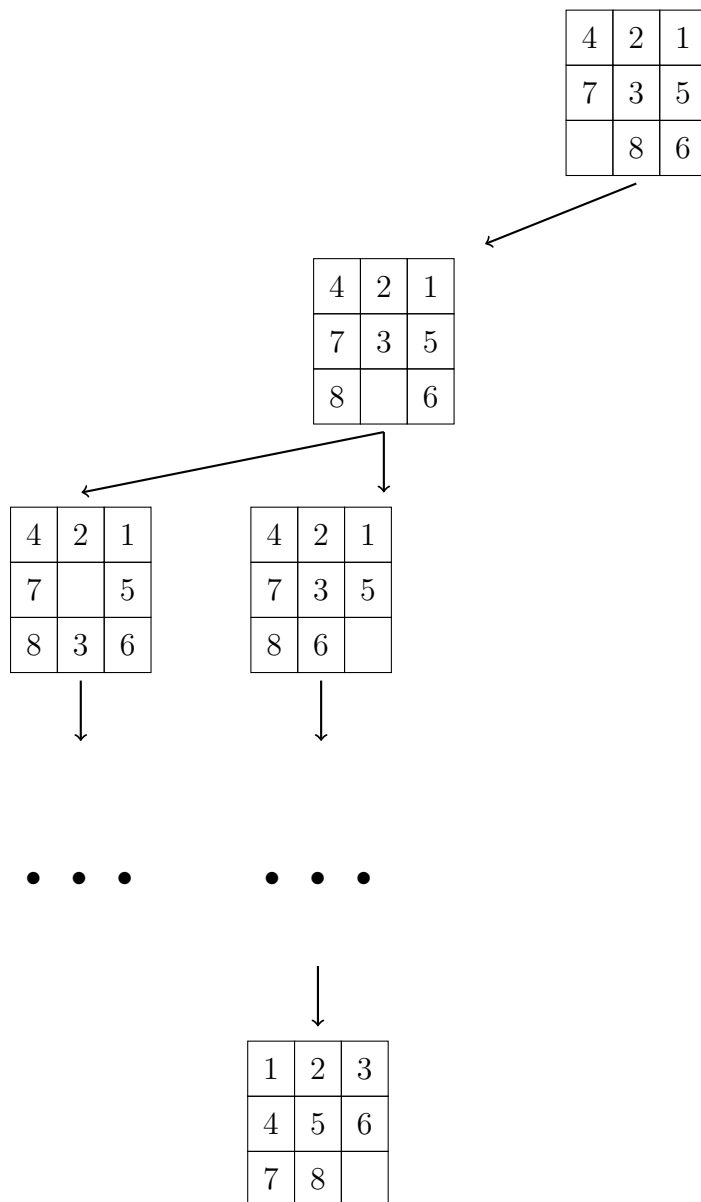
En este método se toma un *estado inicial*. Posteriormente se realiza una llamada recursiva con sus vecinos y en cada llamada se realiza el siguiente proceso:

- $u$  es el estado actual en la llamada recursiva.
- Si  $u$  es igual al *estado final*, terminar.
- Al revisar cada vecino  $v$  si no ha sido visitado, realizar una llamada recursiva en  $v$ .

El método anterior recorrerá los estados siguiendo el primer vecino que no ha sido visitado para cada estado  $u$ , después seguirá con el segundo pero hasta después de no poder avanzar mas en el camino del primer vecino. Eventualmente para cada nodo  $u$  se recorrerán todos sus vecinos hasta llegar al *estado final*.

Visto en una gráfica con estados como vértices es similar al camino mostrado al inicio, sin embargo pudo haber recorrido algún camino que no lo llevo al *estado final*. Cabe mencionar que el recorrido no cumple con algún orden en especial, es decir no se puede determinar el camino mas corto a la solución o hacer alguna clasificación por niveles, a diferencia de las propiedades de realizar una búsqueda en amplitud.

Gráfica búsqueda en profundidad.



### 3. Búsqueda por Ascensión de colinas (Hill Climbing).

En este método se toma un *estado inicial*. Puede usarse el orden de recorrido de la búsqueda de amplitud o profundidad. Supongamos el orden de profundidad, el cual consiste en realizar una llamada recursiva con el *estado inicial*. En cada llamada recursiva se realizan las siguientes operaciones.

- $u$  es el estado actual en la llamada recursiva.
- Si  $u$  es el *estado final*, terminar.
- Se evalúa  $u$  en la función objetivo  $f$  para obtener el valor  $f(u)$ .
- Al revisar cada vecino  $v$ , se evalúa  $v$  en la función objetivo  $f$  para obtener el valor  $f(v)$ .
- Si se cumple que  $f(v) > f(u)$  realizar una llamada recursiva en  $v$ .
- Si no hay tal vecino  $v$ , termina.

El método anterior recorrerá los estados siguiendo el camino que estrictamente aumente el valor de la función  $f$ , si no tiene algún vecino que no tenga un valor mayor al ser evaluado se detiene. Esto hace que a diferencia de los métodos anteriores es posible que nunca llegue al *estado final*. Sin embargo, dependerá de lo adecuada que sea la función objetivo.

Una función objetivo puede ser una que al evaluarla devuelva el número de piezas que estén en su posición correcta con respecto al *estado final*.

$f(u) = \# \text{piezas en posición con respecto al estado final.}$

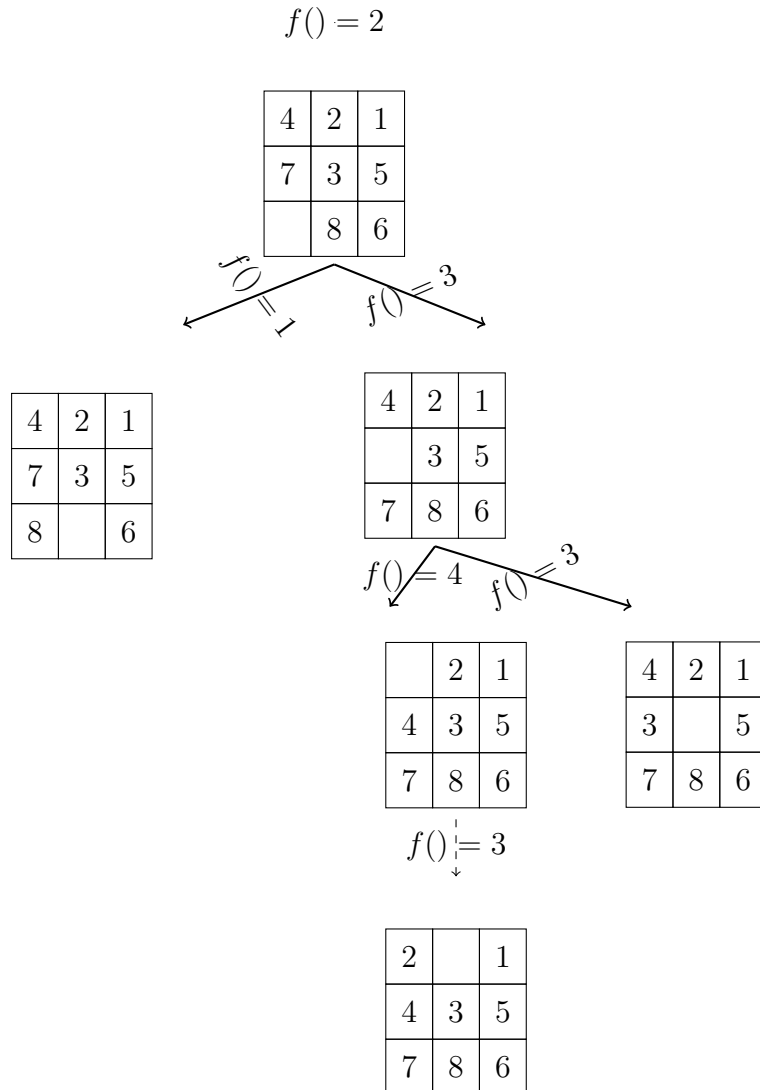
Así evaluando a:

4	2	1
7	3	5
	8	6

Obtenemos 2 por que las piezas 2 y 8 están en la misma posición que en el *estado final*. Visto en una gráfica con estados como vértices es parecida a la gráfica de la búsqueda en profundidad.



## Búsqueda Hill Climbing



El estado final es el que tiene  $f() = 4$  puesto que la única transición a partir de ese estado es la que está con la línea punteada y no la tomará pues tiene un valor menor. Aquí podemos observar que no se llegó al *estado final*.