El 8-puzle consiste en deslizar las fichas horizontalmente o verticalmente al espacio vacío hasta obtener la configuración deseada. Por ejemplo:

Di queremos obtener la siguiente configuración:

1	2	3
4	5	6
7	8	

A partir de el siguiente estado inicial:

1	2	3
	4	6
7	5	8

Se deberían seguir los siguientes pasos:

1. Mover 4 al espacio en blanco

1	2	3
4		6
7	5	8

2. Mover 5 al espacio en blanco

1	2	3
4	**5**	6
7		8

3. mover 8 al espacio en blanco

1	2	3
4	5	6
7	**8**	

Un 15-puzle sigue la misma lógica pero con más fichas, por ejemplo:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Escriba un programa en python3 que resuelva el problema del 15-puzle con las técnicas de búsqueda voraz primero el mejor y la búsqueda A*

Para la búsqueda voraz primero el mejor utilice la heurística "**Distancia de Hamming**":

- h_1 = número de piezas mal colocadas, también llamada distancia de Hamming. Para el ejemplo del 8-puzle anterior,las fichas 4, 5 y 8 están fuera de su posición, así que el espacio inicial tiene $h_1 = 3$. h_1 es admisible, porque está claro que cualquier pieza que está fuera de su lugar debe moverse por lo menos una vez.

Para la búsqueda A* utilice la heurística "**Distancia de Manhattan**"

- h_2 = suma de las distancias de las piezas hacia sus posiciones objetivo. Como las piezas no pueden moverse en diagonal, la distancia que contaremos será la suma de las distancias horizontales y verticales. Esto se llama distancia en la ciudad o distancia Manhattan. h_2 es también admisible, porque cualquier movimiento que se pueda hacer es mover una pieza un paso más cerca del objetivo. Las piezas 1 a 8 en el estado inicial del ejemplo del 8-puzle nos da una distancia de Manhattan de:

Ficha	1	2	3	4	5	6	7	8	
h_2 =	0 +	0 +	0 +	1 +	1 +	0 +	0 +	1 =	3

Si el estado inicial fuera:

3	1	5
7		8
4	6	2

La distancia de Manhattan sería:

Ficha	1	2	3	4	5	6	7	8	
h_2 =	1 +	3 +	2 +	1 +	2 +	2 +	1 +	2 =	14

El 8-puzle o 15-puzle lo puede representar con una lista de listas por ejemplo el 8-puzle del inicio se puede ver en python como:

puzle = [[1,2,3],[0,4,6],[7,5,8]]

Note que la casilla en blanco esta representada con un cero

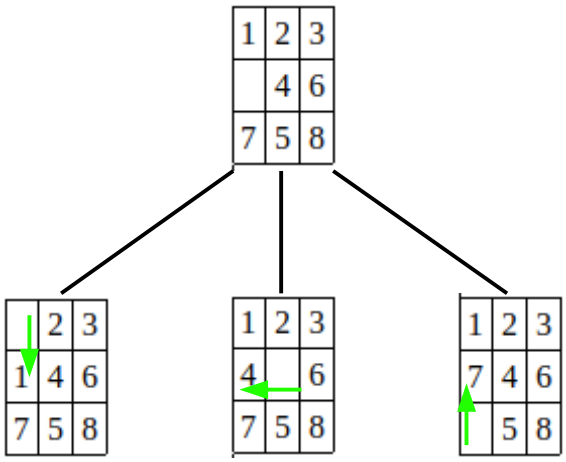
Para saber que fichas se pueden mover hacia la casilla vaciá basta con revisar las posiciones. Y así calcular que fichas en posiciones contiguas se pueden mover hacia la casilla en blanco. Por ejemplo:

1	2	3
	4	6
7	5	8

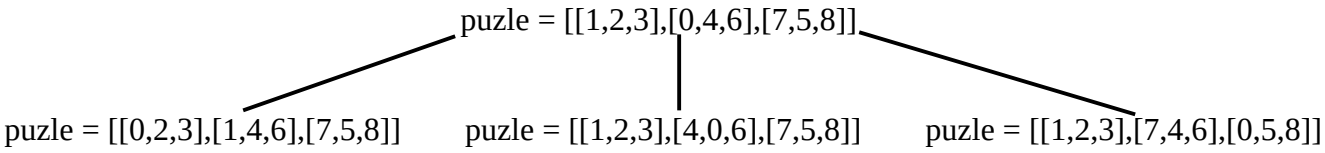
Representado como puzle = [[1,2,3],[0,4,6],[7,5,8]]

Podemos ver que la posición en blanco (el cero) esta en la posición puzle[1][0], las fichas que podemos mover hacia el espacio en blanco son el 1, 4 y 7. Es decir las fichas en posición puzle[0][0], puzle[1][1] y puzle[2][0]. Los indices de la casilla vacía [1][0] se incrementan o decrementan en 1 para obtener las fichas que se pueden mover. Solo debemos tener cuidado con incrementar o decrementar el indice y que éste se salga de los límites del puzle, por ejemplo si el indice es 0 y lo decrementamos quedaría en -1 y es claro que no podemos acceder a esta posición en la lista de listas.

Por lo tanto del estado inicial, los posibles estados siguientes son:



O visto desde la lista de listas:



Como se puede apreciar del ejemplo anterior para generar los siguientes movimientos posibles basta con intercambiar los valores de las posiciones mencionadas más arriba.

Nota. Para el algoritmo A* utilicen el costo de cada movimiento en 1 es decir g(n) es la suma de todos los movimientos hechos hasta n.