**Laboratorio Nº1 resolución problema 8 puzle**

**Profesor:**Matt Bardeen

**Autor:** Matias erenchun

**Introducción**

En el presente informe se expondrá los resultados obtenidos al aplicar la técnica de búsqueda en anchura y búsqueda en profundidad para la resolución del juego 8 puzle.

Para cada una de las técnicas aplicadas se detallará parte del algoritmo, las partes más importantes de su implementación y los resultados obtenidos en dos casos de pruebas de distinto nivel de dificultad, primero se expondrá los resultados de dichas pruebas y al finalizar se comprarán los resultados, además de mencionar consideraciones u comportamientos observados.

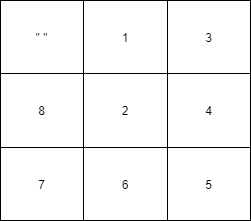
**Nota:** al final del informe se encuentra información sobre el uso del programa.

**Desarrollo**

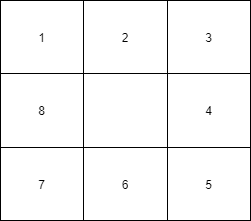
**Casos de prueba**

**Caso 1**

**Estado inicial.**

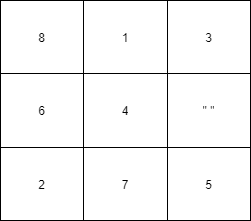


**Estado Final.**

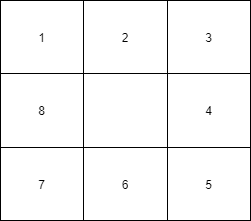


**Caso 2**

**Estado inicial**



**Estado final**



**Planteamiento de la solución.**

Se planteo el problema como un espacio el cual se mueve por el tablero intercambiando su posición con las otras fichas (los números ), de tal manera que en el peor caso ( cuando está en el centro del tablero ) puede generar 4 nuevos estados y en el mejor caso (cuando el espacio esta en las esquinas ) tan solo puede generar 2 nuevos estados, de esta forma se busca de manera se busca en cada iteración cuales son los posibles movimientos que puede generar el estado actual y se agregan a la estructura de almacenamiento de cada tipo de búsqueda( pila u cola según corresponda)

**Búsqueda en anchura**

La búsqueda en anchura se implementó basándose en una cola, de esta manera los nodos se agregaban al final de la cola, por ende, primero se revisan todos los nodos/estados correspondientes al mismo nivel antes de pasar a los del nivel siguiente hasta encontrar el estado de fin.

Para la búsqueda en anchura en el caso 1 se obtuvo el siguiente resultado

nodos expandidos totales:19

niveles expandidos:3

**Resultados búsqueda en anchura caso 2**

Nodos expandidos totales: 32622

Niveles expandidos: 10

**Búsqueda en profundidad**

La búsqueda en profundidad se implemento mediante el uso de una pila y la búsqueda se hizo en sentido de las manecillas del reloj, además para cada nodo revisado se generan y cargan todos los nodos hijos en la pila, pero no se revisan todos los nodos hijos solo el que entro ultimo y este de no ser el estado final generara y agregara nuevos hijos.

Para evitar la entrada en bucles cortos no se consideran los estados en los cuales el estado anterior de un nodo y el estado hijo son iguales.

Para la búsqueda en profundidad en el caso 1 uno se obtuvieron los siguientes resultados:

Nodos expandidos: 52

Niveles expandidos :28

Resultados búsqueda en profundidad caso 2:

Se detuvo el experimento al alcanzar estos resultados

nivel del árbol en que se encuentra este estado:1035076

id estado padre:1897638

id estado actual:1897639

podemos observar que el árbol tenía más de 1 millón de niveles y más de 1,8 millones de nodos/estados.

**Consideraciones, Análisis y conclusiones**

Un punto importante a tener en cuenta es que la búsqueda en profundidad es altamente sensible a la forma de cómo se elige el orden de análisis de los posibles estados hijos, un ejemplo claro es que en el primer caso si la búsqueda era en contra de las manecillas del reloj para poder encontrar un resultado tomaba mas de 15 mil nodos y niveles.

Continuando con lo anterior podemos decir también que la búsqueda en profundidad tiene el problema de entrar en bucles de búsqueda lo cual la deja entrampada hasta el punto de no poder encontrar la solución.

Por otro lado, la búsqueda en anchura para estos casos donde es sabido que la respuesta esta en niveles superficiales, el nivel 9 esta en la superficie en comparación con el nivel 1 millón, resulta bastante buena lo que no quita la gran cantidad de nodos que se deben almacenar en apenas 10 niveles.

Un aprendizaje aplicable en ambos tipos de búsqueda es que se puede mejorar su funcionamiento si se cuenta con información complementaria sobre la ubicación de la respuesta.

**Uso del programa**

El programa esta escrito en java 1.8 usando intellij IDEA

Al ejecutarlo desplegará un menú en la consola donde dará opción de ejecutar las búsquedas.

Si desea cambiar el caso con el que esta cargado por defecto el programa (es el caso de pruebas 2) deberá en la clase main cambiar manual mente dos secciones del código:

Para la matriz de inicio las líneas 14,15 y 16 de la clase main

String [][] inicio= {{"8","1","3"}  
 ,{"6","4"," "}  
 ,{"2","7","5"}};

Recuerde dejar el espacio y en las línea 29 de la clase main

Estado estadoInicial = new Estado( 1,2,inicio,0,0);

Cambiar el primer valor en el constructor de Estado por la fila en donde se encuentre el espacio y el segundo valor por el de la columna donde se encuentre el estado del caos de pruebas nuevo