

UNIVERSIDAD DE LOS LAGOS

INTELIGENCIA ARTIFICIAL

Primer Parcial

Autor:

Diego Muñoz
Cristian Oyarzo
Victor Rodriguez
Sebastian Vidal

Profesor:

Joel Torres

14 de noviembre de 2020



INGENIERÍA CIVIL EN
INFORMÁTICA
UNIVERSIDAD DE LOS LAGOS

Índice

1. Representación	2
2. Algoritmo de búsqueda	2
2.1. Heurística	2
3. Ejemplos	3
4. Solucion	4
5. Observaciones	4

1. Representación

Consideraremos un arreglo de largo 7 con 1s y 2s con un pivote en el medio representado por un 0. La condición inicial es la siguiente:

```
[1, 1, 1, 0, 2, 2, 2]
```

La búsqueda termina si el pivote no tiene lugar hábil de movimiento o llega al siguiente estado final:

```
# Estado Final
```

```
[2, 2, 2, 0, 1, 1, 1]
```

```
# Ejemplo de estado sin movimiento habil siguiente
```

```
[1, 1, 1, 2, 2, 2, 0]
```

Para el movimiento, el pivote puede intercambiar lugar con celdas contiguas si estas cumplen con las siguientes condiciones:

- Movimientos permitidos:
 - “l” movimiento a la izquierda.
 - “ll” dos movimientos a la izquierda.
 - “r” movimiento a la derecha.
 - “rr” dos movimientos a la derecha.
- Ningún movimiento puede dejar que el pivote salga del arreglo.
- El pivote puede moverse hacia la izquierda solo si en esa posición hay 1s.
- El pivote puede moverse hacia la derecha solo si en esa posición hay 2s.

2. Algoritmo de busqueda

Para la busqueda de soluciones al problema se utilizaron los siguientes algoritmos:

- Busqueda no informada: Por profundidad.
- Busqueda informada: Primero el mejor.

2.1. Heuristica

Para la heuristica se puntua el estado del nodo actual en relacion a cuantos valores coinciden al estado final.

```
# Ejemplo de puntuacion de un estado.
```

```
[1, 0, 2, 1, 1, 2, 2]
    +1    +1          => score = 2
[2, 2, 2, 0, 1, 1, 1]
```

3. Ejemplos

Tree print:

```
[1, 1, 1, 0, 2, 2, 2]
  [1, 1, 0, 1, 2, 2, 2]
    [1, 0, 1, 1, 2, 2, 2]
      [0, 1, 1, 1, 2, 2, 2]
        [0, 1, 1, 1, 2, 2, 2]
          [1, 1, 2, 1, 0, 2, 2]
            [1, 1, 2, 0, 1, 2, 2]
              [1, 0, 2, 1, 1, 2, 2]
                [0, 1, 2, 1, 1, 2, 2]
                  [2, 1, 0, 1, 1, 2, 2]
                    [2, 0, 1, 1, 1, 2, 2]
                      [1, 2, 0, 1, 1, 2, 2]
                        [0, 2, 1, 1, 1, 2, 2]
                          [2, 0, 1, 1, 1, 2, 2]
                            [1, 1, 2, 2, 1, 0, 2]
                              [1, 1, 2, 2, 0, 1, 2]
                                [1, 1, 2, 2, 2, 1, 0]
                                  [1, 1, 2, 2, 2, 0, 1]
                                    [1, 1, 2, 2, 1, 2, 0]
                                      [1, 1, 2, 2, 0, 2, 1]
                                        [1, 1, 2, 2, 2, 0, 1]
                                          [1, 1, 2, 1, 2, 0, 2]
                                            [1, 1, 2, 0, 2, 1, 2]
                                              [1, 0, 2, 1, 2, 1, 2]
                                                [0, 1, 2, 1, 2, 1, 2]
                                                  [2, 1, 0, 1, 2, 1, 2]
                                                    [2, 0, 1, 1, 2, 1, 2]
                                                      [2, 1, 2, 1, 0, 1, 2]
                                                        [2, 1, 2, 0, 1, 1, 2]
                                                          [2, 0, 2, 1, 1, 1, 2]
                                                            [2, 2, 0, 1, 1, 1, 2]
                                                              [2, 1, 2, 1, 2, 1, 0]
                                                                [2, 1, 2, 1, 2, 0, 1]
                                                                  [2, 1, 2, 0, 2, 1, 1]
                                                                    [2, 0, 2, 1, 2, 1, 1]
                                                                      [2, 2, 0, 1, 2, 1, 1]
                                                                        [2, 2, 2, 1, 0, 1, 1]
                                                                          [2, 2, 2, 0, 1, 1, 1]
                                                                            [2, 1, 2, 2, 0, 1, 1]
                                                                              [1, 2, 0, 1, 2, 1, 2]
                                                                                [1, 1, 2, 2, 0, 1, 2]
                                                                                  [1, 1, 2, 1, 2, 2, 0]
                                                                                    [1, 1, 2, 1, 2, 2, 0]
                                                                                      [1, 0, 1, 1, 2, 2, 2]
                                                                                        [1, 1, 1, 2, 0, 2, 2]
                                                                                          [1, 1, 1, 2, 2, 0, 2]
```

4. Solucion

Game board initial state:

[1, 1, 1, 0, 2, 2, 2]

Iterations: 34

[1, 1, 1, 0, 2, 2, 2]
[1, 1, 0, 1, 2, 2, 2]
[1, 1, 2, 1, 0, 2, 2]
[1, 1, 2, 1, 2, 0, 2]
[1, 1, 2, 0, 2, 1, 2]
[1, 0, 2, 1, 2, 1, 2]
[0, 1, 2, 1, 2, 1, 2]
[2, 1, 0, 1, 2, 1, 2]
[2, 1, 2, 1, 0, 1, 2]
[2, 1, 2, 1, 2, 1, 0]
[2, 1, 2, 1, 2, 0, 1]
[2, 1, 2, 0, 2, 1, 1]
[2, 0, 2, 1, 2, 1, 1]
[2, 2, 0, 1, 2, 1, 1]
[2, 2, 2, 1, 0, 1, 1]
[2, 2, 2, 0, 1, 1, 1]

5. Observaciones

Durante el desarrollo se implementó la búsqueda por anchura como búsqueda no informada, cambiada posteriormente a búsqueda por profundidad dado que la simetría del problema desarrollaba en anchura un árbol extenso, en cambio en profundidad este se reducía a menos de la mitad.

Comparando los resultados entre ambas búsquedas implementadas (no informada e informada), se pudo observar que la búsqueda informada realizó menos iteraciones que la no informada (34 vs 38 iteraciones).