

LAPORAN TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA
PENYELESAIAN IQ PUZZLE PRO dengan ALGORITMA BRUTE
FORCE



Oleh: Julius Arthur (13523030)

SEKOLAH TEKNIK ELEKTRO DAN ELEKTRONIKA
INSTITUT TEKNOLOGI BANDUNG

2024/2025

BAB I

PENDAHULUAN

IQ puzzle pro adalah permainan papan yang dipopulerkan oleh Smart Games. Dalam permainan ini, pemain harus mengisi sebuah papan dengan blok/ piece yang tersedia. Setiap piece adalah unik satu sama lain. Piece yang tersedia dapat mengalami rotasi dan pencerminan. Permainan ini dinyatakan berhasil jika seluruh papan dapat terisi dengan piece yang tersedia dan semua piece digunakan tanpa terkecuali.

Pada program ini, akan digunakan algoritma brute force untuk mencari sebuah solusi yang mungkin dari permainan IQ puzzle pro. Program dimulai dengan sebuah papan kosong dan mencoba berbagai kemungkinan berdasarkan piece yang tersedia.

BAB II

ALGORITMA BRUTE FORCE

Brute force adalah sebuah algoritma yang mencoba berbagai kemungkinan yang ada hingga ditemukan sebuah solusi. Pada permainan ini, brute force diterapkan pada cara peletakan piece pada papan. Setiap piece memiliki beberapa orientasi akibat adanya rotasi dan pencerminan. Pada program ini, papan dan piece direpresentasikan sebagai sebuah matriks persegi/ persegi panjang.

Langkah – langkah program:

1. Program menyimpan seluruh piece yang tersedia dalam sebuah array.
2. Program memulai brute force sesuai urutan piece yang disimpan pada array. Pada suatu piece, program akan menyimpan seluruh kemungkinan orientasi. Hal ini didapat dengan merotasi piece sebanyak 3 kali ke kanan, dan mencerminkan piece yang ada serta 3 buah hasil rotasinya pada sumbu horizontal dan vertikal. Akibatnya, terdapat 11 piece baru untuk setiap piece.
3. Program melakukan iterasi pada setiap orientasi piece yang terpilih. Pada piece yang terpilih, program mencoba meletakkan piece pada papan dengan posisi (x, y) dengan x dan y adalah koordinat pojok kiri atas dari matriks piece pada matriks papan. Jika ditemukan sebuah posisi yang memungkinkan (tidak tumpang tindih dengan piece yang lain dan dapat diletakkan pada papan), program akan memanggil fungsi solve kembali dengan piece yang baru.
4. Namun, jika tidak ditemukan posisi yang sesuai pada setiap orientasi suatu piece, program akan melakukan backtrack pada proses yang sudah berjalan. Backtrack dilakukan dengan menghapus piece sebelumnya dari papan. Program akan mundur satu langkah dan mencoba meletakkan kembali piece yang dihapus dengan posisi atau orientasi yang berbeda. Program akan kembali pada kondisi no. 3.
5. Jika seluruh piece sudah digunakan dan setiap posisi pada papan sudah diisi oleh piece, maka program telah menemukan sebuah solusi dan menghentikan algoritma brute force. Jika program sudah melakukan iterasi pada setiap piece, orientasi piece, serta posisi, dan tidak dapat mengisi papan dengan penuh, maka permainan ini dinyatakan tidak memiliki solusi.

BAB III

SOURCE CODE

Akan ditampilkan source code algoritma brute force yang digunakan.

```
1 package Tucil1_13523030.src;
2
3 import java.util.concurrent.atomic.AtomicInteger;
4 public class Game
5 {
6     public Board board;
7     public Piece[] pieces;
8
9     public Game(Board b1, Piece[] p){
10         this.board = b1;
11         this.pieces = new Piece[b1.puzzle_pieces];
12         this.pieces = p;
13     }
14
15     public boolean solve_game(int index){
16         if (index == board.puzzle_pieces && board.check()){
17             board.cetakBoard();
18             return true;
19         }
20         Piece[] pieces_rotate = new Piece[12];
21         pieces_rotate[0] = pieces[index];
22         pieces_rotate[1] = pieces[index].mirror_sb_horizontal();
23         pieces_rotate[2] = pieces[index].mirror_sb_vertical();
24
25         int j = 0;
26         int i = 3;
27         while (i < 12){
28             pieces_rotate[i] = pieces_rotate[j].rotate();
29             pieces_rotate[i + 1] = pieces_rotate[j].mirror_sb_horizontal();
30             pieces_rotate[i + 2] = pieces_rotate[j].mirror_sb_vertical();
31
32             j = j + 3;
33             i = i + 3;
34         }
35
36         for (int k = 0; k < 12; k++){
37             for (int x = 0; x < board.M; x++){
38                 for (int y = 0; y < board.N; y++){
39                     // System.out.println(Main.count);
40                     // Main.count.incrementAndGet();
41                     if (board.setPiece(pieces_rotate[k], x, y)){
42                         if (solve_game(index + 1)){
43                             return true;
44                         }
45                     }
46                     board.removePiece(pieces_rotate[k], x, y);
47                 }
48             }
49         }
50
51         return false;
52     }
53 }
54
55
```

BAB I

TEST

Akan ditampilkan 7 contoh input dan output.

1.

```
1 5 5 7
2 DEFAULT
3 A
4 AA
5 B
6 BB
7 C
8 CC
9 D
10 DD
11 EE
12 EE
13 E
14 FF
15 FF
16 F
17 GGG
```

Nama file: tes/game.txt
Game Start
ADDEE
AADEE
BFFFE
BBFFC
GGGCC
Waktu pencarian: 540 ms
Banyak kasus yang ditinjau: 2601383
Apakah anda ingin menyimpan solusi? (ya/tidak)ya

```

1  5 7 9
2  DEFAULT
3  AAA
4  |  AA
5  BBB
6  |  B
7  |  B
8  CC
9  DD
10 DD
11 |  D
12 |  E
13 EEE
14 |  E
15 F
16 G G
17 GGG
18 |  HH
19 HH
20 |  I
21 II

```

2.

```

Nama file: tes/game.txt
Game Start
AAACCHH
IIAAHHE
IBBBEEE
FDDBGEG
DDDBGGG
Waktu pencarian: 848317 ms
Banyak kasus yang ditinjau: 468693701
Apakah anda ingin menyimpan solusi? (ya/tidak)ya

```

```

1  5 5 6
2  DEFAULT
3  AAA
4  √ A
5  |  BB
6  BB
7  B
8  CC
9  √ CC
10 |  DD
11 √ DD
12 |  E
13 EEEE
14 FFF

```

3.

```

Nama file: tes/game.txt
Game Start
AAACC
ABBCC
BBDDF
BDDFF
EEEF
Waktu pencarian: 17 ms
Banyak kasus yang ditinjau: 12729
Apakah anda ingin menyimpan solusi? (ya/tidak)ya

```

4.

```

1  5 5 6
2  DEFAULT
3  AA
4  A
5  | B
6  BB
7  | BB
8  C
9  CC
10 DDD
11 D D
12 | F
13 FF
14 | F
15 | G
16 | G
17 GGG

```

```

Nama file: tes/game.txt
Game Start
AADDD
ABDDF
BBFFG
CBBFG
CCGGG
Waktu pencarian: 14 ms
Banyak kasus yang ditinjau: 43
Apakah anda ingin menyimpan solusi? (ya/tidak)ya

```

```

1  5 4 6
2  DEFAULT
3  A
4  AA
5  A
6  BBB
7  | BB
8  D
9  | E
10 EEEE
11 FFFF
12 | F
13 GGGG
14 GG

```

5.

```

Nama file: tes/game.txt
Game Start
Waktu pencarian: 30220 ms
Banyak kasus yang ditinjau: 301098480
No possible solution

```

```

1  5 5 6
2  DEFAULT
3  A
4  AA
5  A
6  ✓ BBB
7  | BB
8  C
9  DDDD
10 ✓ DD
11 | E
12 EEEE
13 ✓ FFF
14 | F

```

6.

```

Nama file: tes/game.txt
Game Start
ABBBBC
AABBE
AEEEE
DDDDF
DDFFF
Waktu pencarian: 287 ms
Banyak kasus yang ditinjau: 1267578

```



```

1  5 11 12
2  DEFAULT
3  AA
4  A
5  A
6  BB
7  ✓ BBB
8  | C
9  CC
10 C
11 D
12 ✓ DD
13 | DD
14 F
15 F
16 ✓ FF
17 | F
18 ✓ GGGG
19 | G
20 ✓ HH
21 | H
22 | H
23 | H
24 I
25 ✓ II
26 | J
27 ✓ JJ
28 | JJ
29 KKK
30 ✓ K K
31 | L
32 ✓ LL
33 ✓ L
34 M
35 M
36 MMM

```

7.

```

Nama file: tes/game.txt
Game Start
AACFGGGGKKK
ACCFHHGJKLK
ACDFHJJLLM
BBDDFHJJLM
BBBDDHIIMMM
Waktu pencarian: 222 ms
Banyak kasus yang ditinjau: 824615
Apakah anda ingin menyimpan solusi? (ya/tidak)tidak

```

Link repositori: https://github.com/julius123123/Tucil1_13523030/tree/main

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi tanpa kesalahan	V	
2	Program berhasil dijalankan	V	
3	Solusi yang diberikan program benar dan mematuhi aturan permainan	V	
4	Program dapat membaca masukan berkas .txt serta menyimpan solusi dalam berkas .txt	V	
5	Program memiliki <i>Graphical User Interface</i> (GUI)		V
6	Program dapat menyimpan solusi dalam bentuk file gambar		V
7	Program dapat menyelesaikan kasus konfigurasi <i>custom</i>		V
8	Program dapat menyelesaikan kasus konfigurasi Piramida (3D)		V
9	Program dibuat oleh saya sendiri	V	