

LAPORAN TUGAS KECIL 1

IF2211 Strategi Algoritma

Penyelesaian IQ Puzzler Pro dengan Algoritma Brute Force



Disusun oleh:

Shanice Feodora Tjahjono

13523097

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132**

2024

1. Deskripsi Masalah dan Algoritma

1.1 IQ Puzzler Pro

IQ Puzzler Pro adalah permainan papan yang diproduksi oleh perusahaan Smart Games. Tujuan dari permainan ini adalah pemain harus dapat mengisi seluruh papan dengan piece (blok puzzle) yang telah tersedia.

Komponen penting dari permainan IQ Puzzler Pro terdiri dari:

1. Board (Papan) – Board merupakan komponen utama yang menjadi tujuan permainan dimana pemain harus mampu mengisi seluruh area papan menggunakan blok-blok yang telah disediakan.
2. Blok/Piece – Blok adalah komponen yang digunakan pemain untuk mengisi papan kosong hingga terisi penuh. Setiap blok memiliki bentuk yang unik dan semua blok harus digunakan untuk menyelesaikan puzzle.

Permainan dimulai dengan papan yang kosong. Pemain dapat meletakkan blok puzzle sedemikian sehingga tidak ada blok yang bertumpang tindih (kecuali dalam kasus 3D). Setiap blok puzzle dapat dirotasikan maupun dicerminkan. Puzzle dinyatakan selesai jika dan hanya jika papan terisi penuh dan seluruh blok puzzle berhasil diletakkan.

Referensi: Spesifikasi Tugas Kecil 1 Strategi Algoritma 2024/2025

1.2 Penerapan Algoritma Brute Force dalam Menyelesaikan IQ Puzzler Pro

Algoritma yang digunakan menggunakan pendekatan *brute force* dengan *backtracking*. Iterasi dilakukan untuk setiap blok, di mana setiap blok ditentukan posisi yang valid, serta modifikasinya berupa hasil transformasi blok tersebut. Jika ada blok yang tidak bisa ditempatkan, dilakukan backtracking, yaitu mengambil blok terakhir yang ditempatkan dan mencari penempatan lainnya. Proses ini dilakukan hingga semua kemungkinan terpenuhi. Jika semua blok telah ditempatkan dan papan sudah penuh tanpa *overlap*, maka dinyatakan terdapat solusi.

2. Source Code Program

File/Class	Source Code Program
------------	---------------------

Board.java

```
public class Board {
    private char[][] board;
    private int N, M;

    // Constructor
    public Board(int N, int M){
        this.N = N;
        this.M = M;
        board = new char[N][M];

        // Fill up empty spaces with '.'
        for (int i = 0; i < N; i++){
            for (int j = 0; j < M; j++){
                board[i][j] = '.';
            }
        }

        // Getter
        public int getN(){
            return N;
        }
        public int getM(){
            return M;
        }
        public char[][] getBoard(){
            return board;
        }
    }
}
```

```
public boolean placeValid(char[][] piece, int row, int col) {
    if (row + piece.length > N || col + piece[0].length > M) {
        // printMatrix(piece);
        return false;
    }
    for (int i = 0; i < piece.length; i++) {
        for (int j = 0; j < piece[0].length; j++) {
            if (piece[i][j] != '.' && board[row + i][col + j] != '.') {
                // printMatrix(piece);
                return false;
            }
        }
    }
    // printMatrix(piece);
    return true;
}

public void placePiece(char[][] piece, int row, int col) {
    if (!placeValid(piece, row, col)) {
        return;
    }
    // printMatrix(piece);
    for (int i = 0; i < piece.length; i++) {
        for (int j = 0; j < piece[0].length; j++) {
            if (row + i < N && col + j < M && piece[i][j] != '.') {
                board[row + i][col + j] = piece[i][j];
            }
        }
    }
}
```

```

    }
    printBoard();
}

public void removePiece(char[][] piece, int row, int col) {
    // printMatrix(piece);
    for (int i = 0; i < piece.length; i++) {
        for (int j = 0; j < piece[0].length; j++) {
            if (row + i < N && col + j < M && piece[i][j] != '.') {
                board[row + i][col + j] = '.';
            }
        }
    }
    printBoard();
}

// Check if the board is full
public boolean isFull(){
    for (int i = 0; i < N; i++){
        for (int j = 0; j < M; j++){
            if (board[i][j] == '.'){
                return false;
            }
        }
    }
    return true;
}

public void printBoard() {

```

```

    public void printBoard() {
        for (char[] row : board) {
            StringBuilder line = new StringBuilder();
            for (char cell : row) {
                if (cell == '.') {
                    line.append(cell).append(str);
                } else {
                    String color = getColor(cell);
                    line.append(color).append(cell).append(str);
                }
            }
            System.out.println(line.toString());
        }
        System.out.println();
    }

    private String getColor(char id){
        switch (id){
            case 'A': return "\u001B[31m";
            case 'B': return "\u001B[32m";
            case 'C': return "\u001B[33m";
            case 'D': return "\u001B[34m";
            case 'E': return "\u001B[35m";
            case 'F': return "\u001B[36m";
            case 'G': return "\u001B[37m";
            case 'H': return "\u001B[91m";
            case 'I': return "\u001B[92m";
            case 'J': return "\u001B[93m";

```

```

            case 'K': return "\u001B[94m";
            case 'L': return "\u001B[95m";
            case 'M': return "\u001B[96m";
            case 'N': return "\u001B[97m";
            case 'O': return "\u001B[30m";
            case 'P': return "\u001B[90m";
            case 'Q': return "\u001B[40m";
            case 'R': return "\u001B[41m";
            case 'S': return "\u001B[42m";
            case 'T': return "\u001B[43m";
            case 'U': return "\u001B[44m";
            case 'V': return "\u001B[45m";
            case 'W': return "\u001B[46m";
            case 'X': return "\u001B[47m";
            case 'Y': return "\u001B[100m";
            case 'Z': return "\u001B[101m";
            default: return "\u001B[0m";
        }
    }
}

```

Piece.java

```
public class Piece {
    private char[][] piece;
    private char id;

    // Constructor
    public Piece(char[][] piece, char id) {
        this.piece = piece;
        this.id = id;
    }

    // Getter of piece
    public char[][] getPiece() {
        int rows = piece.length, cols = piece[0].length;
        char[][] copy = new char[rows][cols];
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                copy[i][j] = piece[i][j];
        // printMatrix(copy);
        return copy;
    }

    // Getter of piece ID
    public char getID() {
        return id;
    }

    // Rotate 90 degrees
    public char[][] rotate90(char[][] inputPiece) {
        int rows = inputPiece.length, cols = inputPiece[0].length;

        char[][] rotated = new char[cols][rows];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                rotated[j][rows - 1 - i] = inputPiece[i][j];
            }
        }
        // printMatrix(rotated);
        return rotated;
    }

    // Flip horizontally
    public char[][] flipHorizontal() {
        int rows = piece.length, cols = piece[0].length;
        char[][] flipped = new char[rows][cols];
        for (int i = 0; i < rows; i++)
            for (int j = 0; j < cols; j++)
                flipped[i][cols - 1 - j] = piece[i][j];
        // printMatrix(flipped);
        return flipped;
    }

    // Rotate after flip
    public char[][] rotateAfterFlip() {
        char[][] flipped = flipHorizontal();
        char[][] rotated = rotate90(flipped);
        // printMatrix(rotated);
        return rotated;
    }
}
```

FileParser.java

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class FileParser {
    public static ParsedInput parseFile(String filename) throws IOException {
        public static ParsedInput parseFile(String filename) throws IOException {
            File file = new File(filename);
            try (BufferedReader br = new BufferedReader(new FileReader(file))) {
                String firstLine = br.readLine();
                if (firstLine == null || firstLine.trim().isEmpty()) {
                    throw new IOException(message:"File is empty or invalid");
                }

                // Read N, M, and P
                String[] parameters = firstLine.split(regex:" ");
                int N = Integer.parseInt(parameters[0]);
                int M = Integer.parseInt(parameters[1]);
                int P = Integer.parseInt(parameters[2]);

                br.readLine(); // always DEFAULT

                List<String> allLines = new ArrayList<>();
                String line;
                while ((line = br.readLine()) != null) {
                    if (!line.trim().isEmpty()) {
                        allLines.add(line);
                    }
                }

                if (allLines.size() % P != 0) {
                    throw new IOException("Invalid format: Number of lines (" + allLines.size() + ") must be a multiple of " + P);
                }

                List<Piece> pieces = new ArrayList<>();
                char id = 'A';
                int linesPerPiece = allLines.size() / P;

                for (int i = 0; i < P; i++) {
                    int start = i * linesPerPiece;
                    int end = start + linesPerPiece;
                    List<String> currentPiecelines = allLines.subList(start, end);
                    char[][] piece = convertToCharMatrix(currentPiecelines);
                    // printMatrix(piece);
                    pieces.add(new Piece(piece, id++));
                }

                if (pieces.size() != P) {
                    throw new IOException("Expected " + P + " pieces, but found " + pieces.size() + ".");
                }
                return new ParsedInput(N, M, P, pieces);
            }
        }

        // Convert List<String> to char[][] with padding
        private static char[][] convertToCharMatrix(List<String> piecelines) {
            int rows = piecelines.size();
            int cols = 0;
            for (String line : piecelines) {
                cols = Math.max(cols, line.length());
            }
            char[][] piece = new char[rows][cols];

            for (int i = 0; i < rows; i++) {
                String line = piecelines.get(i);
                for (int j = 0; j < cols; j++) {
                    if (j < line.length()) {
                        piece[i][j] = line.charAt(j);
                    } else {
                        piece[i][j] = '.';
                    }
                }
            }
            return piece;
        }
    }

    class ParsedInput {
        public int N, M, P;
        public List<Piece> pieces;

        public ParsedInput(int N, int M, int P, List<Piece> pieces) {
            this.N = N;
        }
    }
}
```

```

class ParsedInput {
    public int N, M, P;
    public List<Piece> pieces;

    public ParsedInput(int N, int M, int P, List<Piece> pieces) {
        this.N = N;
        this.M = M;
        this.P = P;
        this.pieces = pieces;
    }
}

```

Solver.java

```

import java.util.ArrayList;
import java.util.List;

public class Solver {
    private Board board;
    private List<Piece> pieces;
    private long startTime;
    private int iterationCount = 0;

    public Solver(int N, int M, List<Piece> pieces) {
        board = new Board(N, M);
        this.pieces = pieces;
    }

    public boolean solveDetails() {
        startTime = System.currentTimeMillis();
        boolean result = solvePuzzle();
        long endTime = System.currentTimeMillis();
        System.out.println("Waktu pencarian: " + (endTime - startTime) + " ms");
        System.out.println("Banyak kasus yang ditinjau: " + iterationCount);
        if (result) {
            System.out.println(x: "Apakah Anda ingin menyimpan hasil ke file? (y/n)");
        }
        return result;
    }

    public boolean solve(int pieceIndex) {
        iterationCount++;

```

```

        if (pieceIndex >= pieces.size()) {
            board.printBoard();
            return board.isFull();
        }

        Piece piece = pieces.get(pieceIndex);

        for (int i = 0; i < board.getN(); i++) {
            for (int j = 0; j < board.getM(); j++) {
                List<char[][]> transformations = new ArrayList<>();

                transformations.add(piece.getPiece());
                // printMatrix(transformations.get(0));

                transformations.add(piece.flipHorizontal());
                // printMatrix(transformations.get(1));

                char[][] rotated90 = piece.getPiece();
                for (int k = 0; k < 3; k++) {
                    rotated90 = piece.rotate90(rotated90);
                    transformations.add(rotated90);
                    // printMatrix(rotated90);
                }

                char[][] flipped = piece.flipHorizontal();
                char[][] flippedRotated90 = flipped;

```

	<pre> for (int k = 0; k < 3; k++) { flippedRotated90 = piece.rotate90(flippedRotated90); transformations.add(flippedRotated90); // printMatrix(flippedRotated90); } for (char[][] transformation : transformations) { // printMatrix(transformation); if (board.placeValid(transformation, i, j)) { board.placePiece(transformation, i, j); if (solve(pieceIndex + 1)) { return true; } board.removePiece(transformation, i, j); } } } return false; } public boolean solvePuzzle() { return solve(pieceIndex:0); }</pre> <pre>public Board getBoard() { return board; }</pre>
Main.java	<pre>import java.io.IOException; import java.util.Scanner; public class Main { Run Debug public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Masukkan file test case (.txt): "); String filename = scanner.nextLine(); try { ParsedInput input = FileParser.parseFile(filename); Solver solver = new Solver(input.N, input.M, input.pieces); if (solver.solveDetails()) { solver.getBoard().printBoard(); System.out.println("Apakah Anda ingin menyimpan hasil ke file? (y/n)"); String response = scanner.nextLine().trim(); if (response.equalsIgnoreCase("Y")) { saveSolution(solver.getBoard()); } } else { System.out.println("Tidak ada solusi yang ditemukan"); } } catch (IOException e) { System.out.println("File tidak ditemukan atau format tidak valid"); } finally { scanner.close(); } } private static void saveSolution(Board board){ try { java.io.PrintWriter writer = new java.io.PrintWriter(fileName:"solution.txt"); for (char[] row : board.getBoard()) { writer.println(new String(row)); } writer.close(); System.out.println("Solusi disimpan"); } catch (IOException e) { System.err.println("Error saving solution: " + e.getMessage()); } } }</pre>

3. Pengujian

Test Case	Output
-----------	--------

<p>2 2 2 DEFAULT AA AA</p>	<pre>PS C:\Users\x1ani\Downloads\Tucil1_13523097> java -cp src Main Masukkan file test case (.txt): test\test1.txt A A . . A A A A A A A A Waktu pencarian: 4 ms Banyak kasus yang ditinjau: 3 Apakah Anda ingin menyimpan hasil ke file? (y/n) A A A A Apakah Anda ingin menyimpan hasil ke file? (y/n)</pre>
<p>2 2 2 DEFAULT A A B B</p>	<pre>PS C:\Users\x1ani\Downloads\Tucil1_13523097> java -cp src Main Masukkan file test case (.txt): test\test2.txt A . A . A B A B A B A B Waktu pencarian: 3 ms Banyak kasus yang ditinjau: 3 Apakah Anda ingin menyimpan hasil ke file? (y/n) A B A B Apakah Anda ingin menyimpan hasil ke file? (y/n)</pre>
<p>2 3 2 DEFAULT A AA B BB</p>	<pre>PS C:\Users\x1ani\Downloads\Tucil1_13523097> java -cp src Main Masukkan file test case (.txt): test\test3.txt A . . A A . A B B A A B A B B A A B Waktu pencarian: 4 ms Banyak kasus yang ditinjau: 3 Apakah Anda ingin menyimpan hasil ke file? (y/n) A B B A A B Apakah Anda ingin menyimpan hasil ke file? (y/n)</pre>

4 4 7
DEFAULT
A
BB
CCC
DDDD
E
FF
GGG

```
PS C:\Users\x1ani\Downloads\Tucill1_13523097> java -cp src Main  
Masukkan file test case (.txt): test\test4.txt
```

```
A . . .  
. . . .  
. . . .  
. . . .  
A B B .  
. . . .  
. . . .  
. . . .  
A B B C  
. . . C  
. . . C  
. . . .  
A B B C  
. . . C  
. . . C  
. . . C  
D D D D  
A B B C  
E . . C  
. . . C  
D D D D
```

```
A B B C  
E F F C  
. . . C  
D D D D  
A B B C  
E F F C  
G G G C  
D D D D  
A B B C  
E F F C  
G G G C  
D D D D
```

```
Waktu pencarian: 16 ms  
Banyak kasus yang ditinjau: 8  
Apakah Anda ingin menyimpan hasil ke file? (y/n)
```

```
A B B C  
E F F C  
G G G C  
D D D D
```

```
Apakah Anda ingin menyimpan hasil ke file? (y/n)
```

☐

```
5 5 8
DEFAULT
AAAA
BBBBB
CCCCC
DDDD
EE
FF
GG
H
```

```
PS C:\Users\x1ani\Downloads\Tucil1_13523097> java -cp src Main
Masukkan file test case (.txt): test\test5.txt
```

The diagram illustrates the assembly of a 4x4 grid from four 2x2 blocks. The blocks are labeled A, B, and C. The final grid is a 4x4 square composed of these blocks.

A	A	A	A
A	A	A	A
A	A	A	A
A	A	A	A

A A A A .

 A A A A B
 B
 B
 B
 B
 A A A A .

 A A A A B
 B
 B
 B
 B
 A A A A .

A	A	A	A	.
B	B	B	B	B
.
.
.
A	A	A	A	.
B	B	B	B	B
C	C	C	C	C
.
.
A	A	A	A	.
B	B	B	B	B
C	C	C	C	C
D	D	D	D	.
.
A	A	A	A	.
B	B	B	B	B
C	C	C	C	C
D	D	D	D	E
.	.	.	.	E
A	A	A	A	.
B	B	B	B	B
C	C	C	C	C
D	D	D	D	E
F	F	.	.	E

	<pre>A A A A . B B B B B C C C C C D D D D E F F G G E A A A A H B B B B B C C C C C D D D D E F F G G E A A A A H B B B B B C C C C C D D D D E F F G G E Waktu pencarian: 30 ms Banyak kasus yang ditinjau: 13 Apakah Anda ingin menyimpan hasil ke file? (y/n) A A A A H B B B B B C C C C C D D D D E F F G G E Apakah Anda ingin menyimpan hasil ke file? (y/n)</pre>
<p>3 4 5 DEFAULT A BB CCC DDDD EE</p>	<pre>PS C:\Users\xlani\Downloads\Tucil1_13523097> java -cp src Main Masukkan file test case (.txt): test\test6.txt A A B B A B B C . . . C . . . C A B B A B B C . . . C . . . C A B B A B B C . . . C . . . C</pre>

```
A B B .
. . . .
. . . .
```

```
A B B C
. . . C
. . . C
```

```
A B B .
. . . .
. . . .
```

```
A B B .
C C C .
. . . .
```

```
A B B .
C C C .
D D D D
```

```
A B B E
C C C E
D D D D
```

```
A B B E
C C C E
D D D D
```

```
Waktu pencarian: 15 ms
Banyak kasus yang ditinjau: 10
Apakah Anda ingin menyimpan hasil ke file? (y/n)
A B B E
C C C E
D D D D
```

```
Apakah Anda ingin menyimpan hasil ke file? (y/n)
```

```
5 7 8
DEFAULT
AAAA
BBB
CCCCCCC
DD
EEE
FFFFFFF
GGGGG
HHHH
```

```
A A A A B B B
C C C C C C C
D D G G G G G
E E E . . . .
F F F F F F F

A A A A B B B
C C C C C C C
D D G G G G G
E E E H H H H
F F F F F F F

A A A A B B B
C C C C C C C
D D G G G G G
E E E H H H H
F F F F F F F

Waktu pencarian: 1918 ms
Banyak kasus yang ditinjau: 1289
Apakah Anda ingin menyimpan hasil ke file? (y/n)
A A A A B B B
C C C C C C C
D D G G G G G
E E E H H H H
F F F F F F F

Apakah Anda ingin menyimpan hasil ke file? (y/n)
```

4. Lampiran

Referensi
<https://www.geeksforgeeks.org/java-io-bufferedreader-class-java/>
<https://www.geeksforgeeks.org/classes-objects-java/>

Link repository GitHub
https://github.com/feodorashanice/Tucil1_13523097.git

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	