

**LAPORAN TUGAS KECIL**  
**IF2211 STRATEGI ALGORITMA**



**Disusun Oleh:**  
**Muhammad Zakkiy (10122074)**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**

**2025**

## Algoritma Brute Force Untuk IQ Puzzle

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.

Algoritma Brute Force untuk IQ Puzzle:

1. Ambil salah satu blok puzzle lalu tempatkan blok puzzle tersebut pada bagian papan yang masih kosong. Periksa apakah blok puzzle tersebut bisa di tempatkan pada sel-sel tersebut.
2. Jika tidak ada pelanggaran, Tempatkan blok puzzle yang lain pada bagian papan masih kosong dan periksa apakah blok puzzle tersebut bisa ditempatkan pada papan.
3. Jika pada pemeriksaan ditemukan pelanggaran, maka ambil blok puzzle lain dan tempatkan blok puzzle pada sel tersebut.
4. Jika pada proses penempatan ditemukan bahwa tidak satupun dari blok puzzle yang tersisa bisa di tempatkan pada papan, maka tinggalkan bagian papan tersebut dalam keadaan kosong, lalu mundur (backtrack) satu langkah dan ganti blok puzzle yang terakhir kali ditempatkan ke papan menjadi blok puzzle yang lain.
5. Ulangi Langkah 1 sampai papan sudah terisi penuh oleh blok-blok puzzle.

## Source Code Dalam Bahasa Java

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.util.HashMap;
import java.util.ArrayList;

public class IQPuzzler {
    private static int N;
    private static int M;
    private static int P;
    private static String type;
    private static int total_cases;

    private static HashMap<String, ArrayList<ArrayList<Integer>>> pieces = new HashMap<>();
    private static ArrayList<ArrayList<String>> board = new ArrayList<>();
    private static ArrayList<String> key = new ArrayList<>();
```

```

static ArrayList<ArrayList<Integer>> rot(ArrayList<ArrayList<Integer>> piece) {
    int y_min = 0;
    ArrayList<ArrayList<Integer>> rot_piece = new ArrayList<>();

    for (int i = 0; i < piece.size(); i++) {
        rot_piece.add(new ArrayList<>());
        rot_piece.get(i).add(piece.get(i).get(1));
        rot_piece.get(i).add(-piece.get(i).get(0));

        if (y_min > rot_piece.get(i).get(1)) {
            y_min = rot_piece.get(i).get(1);
        }
    }

    ArrayList<ArrayList<Integer>> rot_piece_fix = new ArrayList<>();

    for (int i = 0; i < rot_piece.size(); i++) {
        rot_piece_fix.add(new ArrayList<>());
        rot_piece_fix.get(i).add(rot_piece.get(i).get(0));
        rot_piece_fix.get(i).add(rot_piece.get(i).get(1) - y_min);
    }
    return rot_piece_fix;
}

static ArrayList<ArrayList<Integer>> ref(ArrayList<ArrayList<Integer>> piece) {
    int y_min = 0;
    ArrayList<ArrayList<Integer>> ref_piece = new ArrayList<>();

    for (int i = 0; i < piece.size(); i++) {
        ref_piece.add(new ArrayList<>());
        ref_piece.get(i).add(piece.get(i).get(0));
        ref_piece.get(i).add(-piece.get(i).get(1));

        if (y_min > ref_piece.get(i).get(1)) {
            y_min = ref_piece.get(i).get(1);
        }
    }

    ArrayList<ArrayList<Integer>> ref_piece_fix = new ArrayList<>();

    for (int i = 0; i < ref_piece.size(); i++) {
        ref_piece_fix.add(new ArrayList<>());
        ref_piece_fix.get(i).add(ref_piece.get(i).get(0));
        ref_piece_fix.get(i).add(ref_piece.get(i).get(1) - y_min);
    }
    return ref_piece_fix;
}

static boolean is_valid_placement(ArrayList<ArrayList<Integer>> piece, int x, int y) {
    for (int i = 0; i < piece.size(); i++) {
        int dx = piece.get(i).get(0);
        int dy = piece.get(i).get(1);

        if (x + dx >= N || y + dy >= M) {
            return false;
        } else if (!board.get(x + dx).get(y + dy).isEmpty()) {
            return false;
        }
    }
    return true;
}

static void place_piece(ArrayList<ArrayList<Integer>> piece, int x, int y, String value) {
    if (!value.isEmpty()) {
        total_cases++;
    }
}

```

```

    }

    int n = piece.size();
    for (int i = 0; i < n; i++) {
        int dx = piece.get(i).get(0);
        int dy = piece.get(i).get(1);

        board.get(x + dx).set(y + dy, value);
    }
}

static boolean solve(int index) {

    if (index >= P) {
        boolean is_board_full = true;
        for (int i = 0; i < N; i++) {
            if (board.get(i).contains("")) {
                is_board_full = false;
                break;
            }
        }
        if (is_board_full) {
            return true;
        }
        return false;
    }

    ArrayList<ArrayList<Integer>> piece = pieces.get(key.get(index));

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            if (is_valid_placement(piece, i, j)) {
                place_piece(piece, i, j, key.get(index));
                if (solve(index + 1)) {
                    return true;
                }
                place_piece(piece, i, j, "");
            }

            for (int k = 0; k < 3; k++) {
                piece = rot(piece);
                if (is_valid_placement(piece, i, j)) {
                    place_piece(piece, i, j, key.get(index));
                    if (solve(index + 1)) {
                        return true;
                    }
                }
                place_piece(piece, i, j, "");
            }
        }

        piece = ref(piece);

        if (is_valid_placement(piece, i, j)) {
            place_piece(piece, i, j, key.get(index));
            if (solve(index + 1)) {
                return true;
            }
        }
        place_piece(piece, i, j, "");
    }

    for (int k = 0; k < 3; k++) {
        piece = rot(piece);
        if (is_valid_placement(piece, i, j)) {
            place_piece(piece, i, j, key.get(index));

```

```

        if (solve(index + 1)) {
            return true;
        }
        place_piece(piece, i, j, "");
    }
}
}
}
return false;
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String file_name;

    System.out.println("Masukkan path file(.txt): ");
    file_name = input.nextLine();
    System.out.println();

    try {

        File file = new File(file_name);
        Scanner fileRead = new Scanner(file);
        String firstLine = fileRead.nextLine();

        String[] values = firstLine.split(" ");
        N = Integer.parseInt(values[0]);
        M = Integer.parseInt(values[1]);
        P = Integer.parseInt(values[2]);

        type = fileRead.nextLine();

        String letter = "5";
        int index_r = 0;
        int index_coor = 0;

        while (fileRead.hasNextLine()) {
            String line = fileRead.nextLine();

            String[] first_letter = line.trim().split("");
            if (!first_letter[0].equals(letter)){
                letter = first_letter[0];
                pieces.put(letter, new ArrayList<>());
                key.add(letter);
                index_r = 0;
                index_coor = 0;
            }

            String[] letters = line.split("");

            for (int i = 0; i < letters.length; i++) {
                if (!letters[i].equals(" ")) {
                    pieces.get(letter).add(new ArrayList<>());
                    pieces.get(letter).get(index_coor).add(index_r);
                    pieces.get(letter).get(index_coor).add(i);
                    index_coor++;
                } else {
                    continue;
                }
            }
            index_r++;
        }

        fileRead.close();
    } catch (FileNotFoundException e) {

```

```

        System.out.println("An error occurred.");
        e.printStackTrace();
    }

    for (int i = 0; i < N; i++) {
        board.add(new ArrayList<>());
        for (int j = 0; j < M; j++) {
            board.get(i).add("");
        }
    }

    long startTime = System.nanoTime();
    if (solve(0)) {
        long endTime = System.nanoTime();
        long duration = endTime - startTime;

        String RESET = "\u001B[0m";

        for (int i = 0; i < N; i++) {
            for (int j = 0; j < M; j++) {
                int code_color = board.get(i).get(j).codePointAt(0) - 65;
                System.out.print("\u001B[38;5;" + code_color + "m" + board.get(i).get(j) +
"\u001B[0m");
            }
            System.out.println();
        }

        System.out.println("Banyak kasus yang ditinjau: " + total_cases + "\n");
        System.out.println("Waktu pencarian: " + (duration / 1_000_000.0) + " ms\n");

        String is_create_file;

        System.out.println("Apakah anda ingin menyimpan solusi? (ya/tidak)");
        is_create_file = input.nextLine();

        if (is_create_file.equals("ya")) {
            try {
                FileWriter myWriter = new FileWriter("solution_" + file_name);
                myWriter.write("Input:\n\n");

                try {
                    File file = new File(file_name);
                    Scanner myReader = new Scanner(file);
                    while (myReader.hasNextLine()) {
                        String data = myReader.nextLine();
                        myWriter.write(data + "\n");
                    }
                    myReader.close();
                } catch (FileNotFoundException e) {
                    System.out.println("An error occurred.");
                    e.printStackTrace();
                }

                myWriter.write("\nOutput:\n\n");

                for (int i = 0; i < N; i++) {
                    for (int j = 0; j < M; j++) {
                        myWriter.write(board.get(i).get(j));
                    }
                    myWriter.write("\n");
                }
                myWriter.write("\n");
                myWriter.write("Banyak kasus yang ditinjau: " + total_cases + "\n\n");
                myWriter.write("Waktu pencarian: " + (duration / 1_000_000.0) + " ms\n");
                myWriter.close();
            }

```

```

        System.out.println("Successfully wrote to the file.");
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }
}
} else {
    long endTime = System.nanoTime();
    long duration = endTime - startTime;

    System.out.println("Tidak ada solusi.\n");
    System.out.println("Banyak kasus yang ditinjau: " + total_cases + "\n");
    System.out.println("Waktu eksekusi: " + (duration / 1_000_000.0) + " ms\n");
}
}
}
}

```

### Contoh Input dan Output Program

No	Input	Output
1	<pre> 5 5 7 DEFAULT A AA B BB C CC D DD EE EE E FF FF F GGG </pre>	<pre> ABBCC AABCD EEEDD EEFFF GGGFF Banyak kasus yang ditinjau: 2469  Waktu pencarian: 498.5585 ms  Apakah anda ingin menyimpan solusi? (ya/tidak) </pre>
2	<pre> 3 3 3 DEFAULT A AA BB C CCC </pre>	<pre> AAB ACB CCC Banyak kasus yang ditinjau: 24  Waktu pencarian: 27.1966 ms  Apakah anda ingin menyimpan solusi? (ya/tidak) </pre>
3		

	<pre>4 5 5 DEFAULT A AA AAA B BB C C CC D D DD EEE</pre>	<pre>ABBCC AABCD AAACD EEEDD Banyak kasus yang ditinjau: 6  Waktu pencarian: 8.1672 ms  Apakah anda ingin menyimpan solusi? (ya/tidak)</pre>
4	<pre>6 4 6 DEFAULT A A AA B BB BB CCCC DDD E EE F FF FF</pre>	<pre>ABBB ABBC AAEC DEEC DFFC DFFF Banyak kasus yang ditinjau: 7535  Waktu pencarian: 1755.5019 ms  Apakah anda ingin menyimpan solusi? (ya/tidak)</pre>
5		



	<pre>4 7 7 DEFAULT A AA B B BB CC DD D DD E E EE EE F F FFF GGG</pre>	<pre>A BBBFFF A A B D C F E E D D C F E E E E G G G Banyak kasus yang ditinjau: 12671  Waktu pencarian: 1612.8807 ms  Apakah anda ingin menyimpan solusi? (ya/tidak)</pre>
6	<pre>5 5 6 DEFAULT AA B BB B CC C CC D DD DD E EEE E F FFF</pre>	<pre>A A D D D B B B D D F B E C C F E E E C F F E C C Banyak kasus yang ditinjau: 9535  Waktu pencarian: 2166.3374 ms  Apakah anda ingin menyimpan solusi? (ya/tidak)</pre>
7		

	<pre> 5 6 7 DEFAULT A AA A B BB C CC CCC D DD D E EEE F FFF GG G GG </pre>		<pre> AGGCC AAGCCD AGGCCD EFFFBD EEEFBB Banyak kasus yang ditinjau: 632186  Waktu pencarian: 54694.1224 ms  Apakah anda ingin menyimpan solusi? (ya/tidak) </pre>
--	--	--	---

## Pranala Repository

[https://github.com/UburUburLembur/Tucil1\\_10122074](https://github.com/UburUburLembur/Tucil1_10122074)