

LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA
Penyelesaian IQ Puzzler Pro dengan
Algoritma Brute Force



Indah Novita Tangdililing
13523047 - K01

Daftar Isi

I. SPESIFIKASI TUGAS.....	3
II. ALGORITMA BRUTE FORCE.....	4
III. SOURCE CODE PROGRAM.....	5
IV. TEST CASE.....	12
1. Tidak ada solusi, dimensi balok melebihi dimensi papan.....	12
2. Ada solusi.....	13
3. Ada solusi.....	14
4. Ada solusi.....	15
5. Ada solusi.....	16
6. Tidak solusi, balok lebih banyak dari papan.....	17
7. Ada solusi, tersimpan pada file.....	18
V. REPOSITORI.....	19

I. SPESIFIKASI TUGAS



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**.

II. ALGORITMA BRUTE FORCE

Sebelum mulai masuk ke dalam penyelesaian, program membaca file .txt yang berisi balok-balok puzzle. Cara kerja pembacaan ini, yaitu dengan membaca dua line teratas yang berisi N, M, P, dan mode. Selanjutnya dengan Scanner, program membaca dan menyimpan huruf teratas yang ada, dan menyimpan koordinatnya pada array khusus balok tersebut. Jika pada baris berikutnya ditemukan huruf yang berbeda, maka array yang berisi koordinat tadi, akan di-add ke array pieces puzzle keseluruhan dan program mulai membaca bagian puzzle yang berikutnya. Struktur data pieces balok ini adalah array of array of array of integer.

Algoritma berikutnya adalah prosedur solve, yang memulai penyelesaian puzzle dengan pertama-tama menyalakan “timer”. Selanjutnya, prosedur solve memanggil fungsi solveHelper yang berfungsi secara rekursi mencoba berbagai kemungkinan posisi suatu balok, dan apabila semua balok dapat disusun di board, akan mengembalikan true. Lalu program menutup “timer” dan mengeluarkan output hasil pencarian solusi ke layar.

Cara kerja solveHelper yaitu mencoba memasukkan sebuah balok mulai dari awal hingga blok terakhir di board sampai blok tersebut muat. Jika blok tidak muat, maka akan digeser ke sebelahnya dan ke bawahnya. Jika tetap tidak muat, balok dirotasi dan dicoba lagi dimasukkan. Jika masih tidak muat, setelah rotasi 4 kali, balok dicerminkan, digeser, dirotasi lagi. Jika masih juga tidak muat setelah semua

kemungkinan untuk balok itu dicoba pada papan, program akan melakukan backtracking ke balok yang sebelumnya dipasang untuk mencopotnya lalu mencoba kasus lain lagi.

Program juga memiliki prosedur-prosedur untuk mengubah posisi balok, seperti rotate, flip, place, dan remove. Juga ada fungsi printBoard untuk menampilkan papan dengan warna-warni, dan ada fungsi StrBuilder untuk menyatukan output menjadi sebuah String yang akan ditulis jika pengguna memutuskan untuk menyimpan hasil pencarian pada sebuah file pada komputer..

III. SOURCE CODE PROGRAM

```
import java.io.*;
import java.util.*;

public class Main {
    static int N, M, P;
    static String gameType;
    static char[][] board;
    static List<List<List<Integer>>> pieces = new ArrayList<>();
    static boolean solved = false;
    static int casesExamined = 0;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Masukkan nama file: ");
        String filePath = scan.nextLine();
        try {
            readFile("../test/" + filePath);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        board = new char[N][M];
        for (int u=0; u<N; u++){
```

```

        for (int v=0; v<M; v++){
            board[u][v] = '.';
        }
    }
    solve();
    scan.close();
}

public static void readFile(String fileName) throws
FileNotFoundException {
    File file = new File(fileName);
    Scanner scanFile = new Scanner(file);
    N = scanFile.nextInt();
    M = scanFile.nextInt();
    P = scanFile.nextInt();
    List<List<Integer>> currPiece = new ArrayList<>();
    char currentPiece = 'A';
    String line = scanFile.nextLine();
    scanFile.nextLine();
    int x = -1;
    while (scanFile.hasNextLine()){
        line = scanFile.nextLine();
        x++;
        for (int i=0; i<line.length(); i++){
            if (line.charAt(i) != ' ' && line.charAt(i) !=
currentPiece) {
                pieces.add(currPiece);
                currPiece = new ArrayList<>();
                currentPiece = line.charAt(i);
                x = 0;
                break;
            }
        }
        for (int i=0; i<line.length(); i++){
            if (line.charAt(i) == currentPiece){
                List<Integer> square = new ArrayList<>();
                square.add(x);
                square.add(i);
                currPiece.add(square);
                square = new ArrayList<>();
            }
        }
    }
}

```

```

        }
    }
}

pieces.add(currPiece);
scanFile.close();
}

public static void rotate(List<List<Integer>> unrotated) {
    for (int i=0; i<unrotated.size(); i++){
        List<Integer> rotating = new ArrayList<>();
        int x = unrotated.get(i).get(0);
        int y = unrotated.get(i).get(1);
        rotating.add(-1*y);
        rotating.add(x);
        unrotated.set(i, rotating);
    }
    adjust(unrotated);
}

public static void flip(List<List<Integer>> unflipped) {
//parameter pieces.get()
    for (int i=0; i<unflipped.size(); i++){
        List<Integer> flipping = new ArrayList<>();
        int x = unflipped.get(i).get(0);
        int y = unflipped.get(i).get(1);
        flipping.add(-1*x);
        flipping.add(y);
        unflipped.set(i, flipping);
    }
    adjust(unflipped);
}

public static void adjust(List<List<Integer>> unadjusted) {
    int xmin = unadjusted.get(0).get(0);
    int ymin = unadjusted.get(0).get(1);
    for (int i=1; i<unadjusted.size(); i++){
        int x = unadjusted.get(i).get(0);
        int y = unadjusted.get(i).get(1);
        if (x<xmin) {
            xmin = x;
        }
    }
}

```

```

        if (y<ymin){
            ymin = y;
        }
    }
    for (int i=0; i<unadjusted.size();i++){
        List<Integer> adjusting = new ArrayList<>();
        adjusting.add(unadjusted.get(i).get(0)-xmin);
        adjusting.add(unadjusted.get(i).get(1)-ymin);
        unadjusted.set(i,adjusting);
    }
}

public static void place(int pieceIdx, int x_move, int y_move){
    for(int k=0; k<pieces.get(pieceIdx).size(); k++){
        board[pieces.get(pieceIdx).get(k).get(0) +
x_move][pieces.get(pieceIdx).get(k).get(1) + y_move] = (char)
('A'+pieceIdx);
    }
}

public static void remove(int pieceIdx){
    for (int u=0; u<N; u++){
        for (int v=0; v<M; v++){
            if (board[u][v] == 'A'+pieceIdx){
                board[u][v] = '.';
            }
        }
    }
}

public static void solve() {
    long startTime = System.currentTimeMillis();
    if (solveHelper(0)) {
        solved = true;
        System.out.println("Solution found.");
    } else {
        System.out.println("No solution found.");
    }
    long endTime = System.currentTimeMillis();
    long duration = endTime - startTime;
}

```



```

        System.out.println("\nSearch time: " + duration + " ms");
        System.out.println("Cases examined: " + casesExamined);
        printBoard();
        saveToFile(solved, duration);
    }

    public static boolean solveHelper(int pieceIdx) {
        if (pieceIdx == pieces.size()) {
            return true;
        }

        for (int rotation = 0; rotation < 8; rotation++) {
            for (int n = 0; n < N; n++) {
                for (int m = 0; m < M; m++) {
                    casesExamined++;
                    if (canPlace(pieceIdx, n, m)) {
                        place(pieceIdx, n, m);
                        if (solveHelper(pieceIdx + 1)) {
                            return true;
                        }
                        remove(pieceIdx);
                    }
                }
            }
            if (rotation == 3) {
                flip(pieces.get(pieceIdx));
            } else {
                rotate(pieces.get(pieceIdx));
            }
        }
        return false;
    }

    public static boolean canPlace(int pieceIdx, int x_move, int
y_move) {
        for (int k = 0; k < pieces.get(pieceIdx).size(); k++) {
            int x = pieces.get(pieceIdx).get(k).get(0) + x_move;
            int y = pieces.get(pieceIdx).get(k).get(1) + y_move;
            if (x >= N || y >= M || board[x][y] != '.') {
                return false;
            }
        }
    }

```

```

        }
    }
    return true;
}

public static void printBoard() {
    String[] colors = {
        "\u001B[31m", // Red
        "\u001B[32m", // Green
        "\u001B[33m", // Yellow
        "\u001B[34m", // Blue
        "\u001B[35m", // Magenta
        "\u001B[36m", // Cyan
    };
    String reset = "\u001B[0m";

    for (int u = 0; u < N; u++) {
        for (int v = 0; v < M; v++) {
            if (board[u][v] == '.') {
                System.out.print(board[u][v]);
            } else {
                int colorIndex = (board[u][v] - 'A') %
colors.length;
                System.out.print(colors[colorIndex] +
board[u][v] + reset);
            }
        }
        System.out.println();
    }
    System.out.println();
}

public static void saveToFile(boolean found, long duration) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Apakah ingin menyimpan? (y/n): ");
    String save = scan.nextLine();
    if (!save.equals("y")) {
        return;
    }
    System.out.println("Masukkan nama file untuk menyimpan: ");
}

```

```

        String fileName = scan.nextLine();
        String result = StrBuild(found, duration);
        try {
            FileWriter writer = new FileWriter("../test/" +
fileName);
            writer.write(result);
            writer.close();
            System.out.println("Hasil berhasil disimpan ke file " +
fileName);
        } catch (IOException e) {
            System.out.println("Terjadi kesalahan saat menyimpan
jawaban.");
            e.printStackTrace();
        }
    }

    public static String StrBuild(boolean found, long duration) {
        StringBuilder result = new StringBuilder();
        if (found) {
            result.append("Solution found.\n");
        } else {
            result.append("Solution not found.\n");
        }
        result.append("Search time: " + duration + " ms\n");
        result.append("Cases examined: " + casesExamined + "\n\n");
        for (int u = 0; u < N; u++) {
            for (int v = 0; v < M; v++) {
                result.append(board[u][v]);
            }
            result.append("\n");
        }
        return result.toString();
    }
}

```

IV. TEST CASE

1. Tidak ada solusi, dimensi balok melebihi dimensi papan.

Input:

```
1      3 3 3
2      DEFAULT
3      AAA
4      BBBBB
5      C
6      |_____
```

Output:

```
Masukkan nama file:
filetesting.txt
No solution found.

Search time: 2 ms
Cases examined: 1800
...
...
...

Apakah ingin menyimpan? (y/n):
n
```

2. Ada solusi

Input:

```
1 5 5 7
2 DEFAULT
3 AAA
4 B
5 B
6 C
7 CC
8 D
9 DD
10 EEE
11 E E
12 E
13 FF
14 F
15 F
16 GGG
17 | G
18 | _____
```

Output:

```
cmd (C:\> , 11 (P:) \ java\ main.java /
Masukkan nama file:
filetesting.txt
Solution found.
```

```
Search time: 67 ms
Cases examined: 1143461
AAABE
DDEBE
FDEEE
FGGGC
FFGCC
```

```
Apakah ingin menyimpan? (y/n):
n
```

3. Ada solusi

Input:

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	E
13	EE
14	E
15	F
16	F
17	FFF
18	

Output:

```
Masukkan nama file:
filetesting.txt
Solution found.
```

```
Search time: 37 ms
Cases examined: 462797
AADD D
ABDE D
BBEE F
CBBE F
CCFF F
```

```
Apakah ingin menyimpan? (y/n).
```

4. Ada solusi

Input:

```
Tucil-1-Stima > test > ≡ filetesting.t
```

```
1 5 4 5
2 DEFAULT
3 AA
4 A
5 A
6 BB
7 BBB
8 | C
9 CC
10 C
11 DDD
12 E
13 EE
14 | E
15
```

Output:

```
Masukkan nama file:
filetesting.txt
Solution found.
```

```
Search time: 3 ms
Cases examined: 7411
```

```
AACD
```

```
ACCD
```

```
ACED
```

```
BBEE
```


```
BBBE
```

```
Apakah ingin menyimpan? (y/n):
```

```
■
```

5. Ada solusi

Input:

Tucil-1-Stima > test >  filete	
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

Output:

```
1ma\src\ ; if ($?) { javac Main.java } ; if ($?) { ;
```

Masukkan nama file:

filetesting.txt

Solution found.

Search time: 182 ms

Cases examined: 5403457

AGGGC

AABCC

EEBBF

EEDFF

EDDFF

Apakah ingin menyimpan? (y/n):

n

6. Tidak solusi, balok lebih banyak dari papan

Input:

1	3 3 4
---	-------

2	DEFAULT
---	---------

3	AA
---	----

4	AAA
---	-----

5	AA
---	----

6	B
---	---

7	C
---	---

8	D
---	---

-	
---	--

Output:

```
\\msd\src\ ; if ($?) { javac Main.java
```

Masukkan nama file:

filetesting.txt

No solution found.

Search time: 11 ms

Cases examined: 83592

...

...

...

Apakah ingin menyimpan? (y/n):

7. Ada solusi, tersimpan pada file

Input:

Tucil-1-Stima > test > filetesting.txt

1	5 7 6
2	DEFAULT
3	A
4	AAA
5	A
6	BB
7	BB
8	B
9	CCCCC
10	CC
11	DDD
12	D DDD
13	E
14	E
15	FFF F
16	FFFFF
17	

Output:

```
PS C:\Users\62812\Stima\Tucil-1-Stima\src> cd "ima\src\" ; if ($?) { javac Main.java } ; if ($?) {  
Masukkan nama file:  
filetesting.txt  
Solution found.  
  
Search time: 334 ms  
Cases examined: 7008357  
FFDDDEE  
FFDADDD  
FFAAABB  
FCCABBB  
FFCCCCC  
  
Apakah ingin menyimpan? (y/n):  
y  
Masukkan nama file untuk menyimpan:  
hasil.txt  
Hasil berhasil disimpan ke file hasil.txt
```

V. REPOSITORY

Program dapat diakses pada github

https://github.com/indahtangdililing/Tucil1_13523047