

## 1. Deskripsi Algoritma

### *Deskripsi garis besar:*

Kita *bruteforce* semua kemungkinan path yang ada yang valid (valid artinya memenuhi persyaratan path yaitu diawali di baris pertama, dimulai dengan gerakan vertikal, gerakan dilakukan secara selang-seling, dan tidak ada pemilihan cell yang sama lebih dari sekali). Untuk setiap path yang valid, kita iterasikan setiap sequence yang diberikan untuk mengecek apakah sequence tersebut berada di dalam pathnya sehingga pada akhir iterasi telah ditemukan total hadiah dari path tersebut. Jika total hadiah lebih besar daripada hadiah terbesar sebelumnya, maka diupdate jawabannya sehingga menjadi yang terbaru.

### *Deskripsi detail:*

1. Kita punya array multidimensi bertipe boolean ( bernama `taken`) yang berukuran sama dengan ukuran boolean untuk menyatakan apakah cell tersebut sudah pernah di visit atau belum. Jika pada suatu saat kita ingin mundur dalam pencarian, maka cell terakhir di toggle menjadi false.
2. Kita punya dynamic array yang menyimpan lokasi cell-cell yang telah dipilih (misalnya `[(1,1), (1, 4), (3, 4), (3, 5)]`) ( bernama `path`). Jika pada suatu saat kita ingin mundur, maka elemen terakhir di array tinggal di hapus/pop.
3. Kita gunakan fungsi rekursif yang menerima 3 parameter: `row` (integer), `col` (integer), `is_vertical` (boolean). Invarian dari fungsi ini adalah pada saat itu cell yang memiliki `row` dan `col` yang bersangkutan sudah dipilih (sehingga ter-mark true di `taken` dan sudah di push lokasinya di `path`) dan sedang memilih cell yang akan menjadi elemen selanjutnya pada arah yang dinyatakan dengan variabel `is_vertical`. Dalam fungsi tersebut dilakukan
  - a. Langkah pertama adalah mengecek apakah `taken` sudah melebihi ukuran maksimum buffer. Jika iya maka fungsi di-return.
  - b. Langkah kedua adalah mengiterasikan setiap sequence yang diberikan untuk menghitung total hadiah yang dimiliki oleh `path` pada saat itu. Jika total hadiah lebih besar dari yang telah ditemukan sebelumnya, maka diupdate jawabannya.
  - c. Langkah ketiga adalah mengiterasi setiap cell dari `row` atau `column` (`row` jika `is_vertical` bernilai true, dan `column` jika `is_vertical` bernilai false) dari cell terakhir yang dipilih dan mengecek apakah sudah dipilih atau belum melalui `taken`. Jika belum diambil, maka cell tersebut di mark true di `taken` dan lokasinya di push ke `path` dan fungsi `search` dipanggil pada cell itu dengan `is_vertical` dengan berbeda dengan sebelumnya. Setelah fungsi `search` selesai, cell tersebut di mark false dan lokasinya di pop sehingga pada iterasi cell-cell selanjutnya tidak bermasalah.

## 2. Source Program

```
// Lihat cara compile yang detail di README karena tidak bisa dicompile
dengan biasa.
#include <iostream>
#include <vector>
#include <utility>
#include <fstream>
#include <sstream>
#include <random>
#include <chrono>
#include <iomanip>
#include <thread>
#include <ctime>

using std::vector;
using std::string;
using std::pair;
using std::cout;
using std::cerr;
using std::cin;
using std::ifstream;
using std::ofstream;
using std::stringstream;
using std::thread;
using std::ref;

struct Sequence {
    vector<int> elements;
    int value;
};

struct Configuration {
    int width, height;
    vector<vector<int>> matrix;
```

```

    int total_rewards;
    vector<Sequence> rewards;
    int maximum_buffer_size;
};

void search(const Configuration &config, const int &row, const int &col,
const bool &vertical, vector<vector<bool>> &taken, vector<int> &buffer,
vector<int> &optimal_token_path, vector<pair<int, int>> &path,
vector<pair<int, int>> &optimal_path, int &max_reward) {
    int cur_prize = 0;
    for (const auto &reward : config.rewards) {
        for (int i = 0, buf_sz = (int)buffer.size(), rew_sz =
(int)reward.elements.size(); i <= buf_sz - rew_sz; ++i) {
            bool valid = true;
            for (int j = i; valid && j < i + rew_sz; ++j) {
                if (buffer[j] != reward.elements[j-i]) {
                    valid = false;
                }
            }
            if (valid) {
                cur_prize += reward.value;
                break;
            }
        }
    }
    if (cur_prize > max_reward) {
        max_reward = cur_prize;
        optimal_token_path = buffer;
        optimal_path = path;
    }
    if (buffer.size() == config.maximum_buffer_size) {
        return;
    }
    if (vertical) {
        for (int i = 0; i < config.height; ++i) {
            if (!taken[i][col]) {
                buffer.push_back(config.matrix[i][col]);
                taken[i][col] = true;
                path.emplace_back(col, i);
            }
        }
    }
}

```

```

        search(config, i, col, false, taken, buffer,
optimal_token_path, path, optimal_path, max_reward);
        path.pop_back();
        taken[i][col] = false;
        buffer.pop_back();
    }
}
} else {
    for (int i = 0; i < config.width; ++i) {
        if (!taken[row][i]) {
            buffer.push_back(config.matrix[row][i]);
            taken[row][i] = true;
            path.emplace_back(i, row);
            search(config, row, i, true, taken, buffer,
optimal_token_path, path, optimal_path, max_reward);
            path.pop_back();
            taken[row][i] = false;
            buffer.pop_back();
        }
    }
}
}
}

```

```

bool read_text_file(Configuration &config) {
    cout <<
        "Format text file:\n\n"

        "buffer_size\n"
        "matrix_width matrix_height\n"
        "matrix (jumlah row sebanyak matrix_height dan jumlah kolom
sebanyak matrix_width)\n"
        "number_of_sequences\n"
        "sequences_1\n"
        "sequences_1_reward\n"
        "sequences_2\n"
        "sequences_2_reward\n"
        "... \n"
        "sequences_n\n"
        "sequences_n_reward\n\n"

```

```

        "Masukkan alamat lengkap file: ";
cout.flush();

string filename;
cin >> filename;
ifstream input_file(filename);

if (!input_file.is_open()) {
    cerr << "File gagal dibuka.\n\n";
    return false;
}

input_file >> config.maximum_buffer_size;
input_file >> config.width >> config.height;
config.matrix.resize(config.height);
for (int i = 0; i < config.height; ++i) {
    config.matrix[i].resize(config.width);
}
for (int i = 0; i < config.height; ++i) {
    for (int j = 0; j < config.width; ++j) {
        char a, b;
        input_file >> a >> b;
        config.matrix[i][j] = 256 * a + b;
    }
}

input_file >> config.total_rewards;
input_file.ignore();
for (int i = 0; i < config.total_rewards; ++i) {
    Sequence s;
    string elements_line;
    getline(input_file, elements_line);

    stringstream ss(elements_line);
    char a, b;
    while (ss >> a >> b) {
        s.elements.push_back(256 * a + b);
    }
    input_file >> s.value;
    input_file.ignore();
    config.rewards.push_back(s);
}

```

```

    }
    return true;
}

void generate_game(Configuration &config) {
    cout <<
        "\nFormat input:\n\n"
        "Jumlah token unik\n"
        "Token-token\n"
        "Ukuran buffer\n"
        "Ukuran matriks\n"
        "Jumlah sekuens\n"
        "Ukuran maksimal sekuens\n\n";
    cout.flush();

    int total_tokens;
    cin >> total_tokens;
    vector<int> tokens(total_tokens);
    for (int i = 0; i < total_tokens; ++i) {
        char a, b;
        cin >> a >> b;
        tokens[i] = 256 * a + b;
    }
    cin >> config.maximum_buffer_size;
    cin >> config.width >> config.height;
    cin >> config.total_rewards;
    int max_sequence_size;
    cin >> max_sequence_size;

    config.matrix.resize(config.height);
    for (int j = 0; j < config.height; ++j) {
        config.matrix[j].resize(config.width);
    }
    std::random_device dev;
    std::mt19937 generator(dev());
    std::uniform_int_distribution<int> distribution_length(2,
max_sequence_size);
    std::uniform_int_distribution<int> distribution_token(0, total_tokens -
1);
    std::uniform_int_distribution<int> distribution_reward(-13, 10);

```

```

    for (int j = 0; j < config.height; ++j) {
        config.matrix[j].resize(config.width);
        for (int i = 0; i < config.width; ++i) {
            config.matrix[j][i] = tokens[distribution_token(generator)];
        }
    }
    for (int i = 0; i < config.total_rewards; ++i) {
        int length = distribution_length(generator);
        Sequence s;
        for (int j = 0; j < length; ++j) {
            s.elements.push_back(tokens[distribution_token(generator)]);
            s.value = distribution_reward(generator);
        }
        config.rewards.push_back(s);
    }

    cout << "Berikut adalah konfigurasi permainan yang dihasilkan secara acak.\n\n";

    for (int i = 0; i < config.height; ++i) {
        for (int j = 0; j < config.width; ++j) {
            int b = (config.matrix[i][j] % 256);
            int a = ((config.matrix[i][j] - b) / 256);
            cout << (char)a << (char)b << ' ';
        }
        cout << '\n';
    }
    cout << '\n';
    for (const auto &reward : config.rewards) {
        for (const auto &i : reward.elements) {
            int b = (i % 256);
            int a = ((i - b) / 256);
            cout << (char)a << (char)b << ' ';
        }
        cout << '\n';
        cout << reward.value << '\n';
    }
    cout << '\n';
}

```

```

void search_first_row(const Configuration &config, int col, int
&max_reward, vector<pair<int, int>> &optimal_path, vector<int>
&optimal_token_path) {
    vector<vector<bool>> taken(config.height, vector<bool>(config.width));
    vector<int> buffer;
    vector<pair<int, int>> path;

    taken[0][col] = true;
    buffer.push_back(config.matrix[0][col]);
    path.emplace_back(col, 0);
    search(config, 0, col, true, taken, buffer, optimal_token_path, path,
optimal_path, max_reward);
    path.pop_back();
    taken[0][col] = false;
    buffer.pop_back();
}

bool save_text_to_file(const string& content, string filename) {
    ofstream file(filename);

    if (file.is_open()) {
        file << content;
        file.close();
        return true;
    } else {
        return false;
    }
}

int main() {
    cout << "Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma
Bruce Force.\n\n";

    while (true) {
        cout <<
            "Pilih metode input:\n"
            "1. Input dengan text file.\n"
            "2. Input dihasilkan secara otomatis.\n\n";
        cout.flush();
    }
}

```



```

int input_type;
cout << " :> ";
cout.flush();
cin >> input_type;

Configuration config;

if (input_type == 1) {
    if (!read_text_file(config)) {
        continue;
    }
} else if (input_type == 2) {
    generate_game(config);
} else {
    cout << "Input tidak valid.\n\n";
    continue;
}

int max_reward = 0;
vector<pair<int, int>> optimal_path;
vector<int> optimal_token_path;

auto start = std::chrono::steady_clock::now();

vector<int> max_rewards(config.width);
vector<vector<pair<int, int>>> optimal_paths(config.width);
vector<vector<int>> optimal_token_paths(config.width);

vector<thread> threads;
for (int col = 0; col < config.width; ++col) {
    threads.push_back(thread(search_first_row, ref(config), col,
ref(max_rewards[col]), ref(optimal_paths[col]),
ref(optimal_token_paths[col])));
}
for (auto &thread : threads) {
    thread.join();
}
for (int col = 0; col < config.width; ++col) {
    if (max_reward < max_rewards[col]) {
        max_reward = max_rewards[col];
    }
}

```

```

    }

    }

    for (int col = 0; col < config.width; ++col) {
        if (max_rewards[col] == max_reward) {
            optimal_path = optimal_paths[col];
            optimal_token_path = optimal_token_paths[col];
            break;
        }
    }

    auto end = std::chrono::steady_clock::now();

    auto duration =
std::chrono::duration_cast<std::chrono::nanoseconds>(end - start).count();

    stringstream response_stream;

    if (max_reward == 0) {
        response_stream << "Tidak ditemukan solusi yang menghasilkan
total hadiah yang positif.\n";
    } else {
        response_stream << max_reward << '\n';
        for (auto &i : optimal_token_path) {
            int b = (i % 256);
            int a = ((i - b) / 256);
            response_stream << (char)a << (char)b << ' ';
        }
        response_stream << '\n';

        for (const auto &pos : optimal_path) {
            response_stream << pos.first + 1;
            response_stream << ", ";
            response_stream << pos.second + 1;
            response_stream << '\n';
        }
        response_stream << '\n';
    }

    response_stream << (double)duration / 1e6 << " ms";

    string response = response_stream.str();
    cout << response;

```

```

        cout.flush();
        cout << "\n\nApakah ingin menyimpan solusi? (y/n) ";
        cout.flush();
        char confirmation;
        cin >> confirmation;
        if (confirmation == 'y' || confirmation == 'Y') {
            string filename = "result.out";
            if (save_text_to_file(response, filename)) {
                cout << "Solusi telah disimpan di " << filename << "\n\n";
            } else {
                cout << "Solusi gagal disimpan dalam sebuah file karena
error.\n\n";
            }
        } else {
            cout << "Solusi tidak disimpan dalam sebuah file.\n\n";
        }
    }
}

```

### 3. Test

Untuk input yang kecil, executable yang dihasilkan oleh g++ di linux cenderung lebih cepat daripada yang dihasilkan clang++ di Windows. Untuk input yang besar, executable yang dihasilkan oleh clang++ di Windows cenderung lebih cepat daripada yang dihasilkan g++ di linux.

#### 1. Tes sample (linux, g++)

**Input:** test/sample.in

```

7
6 6
7A 55 E9 E9 1C 55
55 7A 1C 7A E9 55
55 1C 1C 55 E9 BD
BD 1C 7A 1C 55 BD
BD 55 BD 7A 1C 1C
1C 55 55 7A 55 7A
3
BD E9 1C
15
BD 7A BD
20
BD 1C BD 55
30

```

**Output:** test/sample.out

```
50
7A BD 7A BD 1C BD 55
1, 1
1, 4
3, 4
3, 5
6, 5
6, 3
1, 3
```

3.00914 ms

## Tangkapan Layar

```
azm@azmicomp:~/projects/Tucil1_13522109$ ./bin/ubuntu_22_04_3
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: /home/azm/projects/Tucil1_13522109/test/sample.in
50
7A BD 7A BD 1C BD 55
1, 1
1, 4
3, 4
3, 5
6, 5
6, 3
1, 3

3.00914 ms

Apakah ingin menyimpan solusi? (y/n) █
```

## 2. Tes matriks besar & sekuens banyak (Windows, clang++)

**Input:** test/1.in

```
6
11 11
E9 E9 E9 7A BD 1C E9 BD 55 7A 1C
1C 7A 7A BD 55 BD 1C E9 55 E9 BD
E9 7A 7A 55 55 E9 BD BD 7A 55 1C
1C 1C BD 55 7A BD E9 7A BD 7A BD
7A 7A E9 55 55 1C 1C 7A 55 55 7A
55 1C 55 55 1C BD E9 55 55 55 E9
BD BD 1C 55 7A 1C 1C 55 E9 BD 1C
BD 1C BD 7A 1C E9 1C BD E9 BD 1C
55 55 7A BD 1C BD E9 E9 1C BD BD
1C BD BD E9 55 55 E9 BD 1C 55 E9
7A 7A 1C 7A BD 55 E9 7A E9 BD 7A
1000
7A 7A
3
1C E9
-7
.....
.....
.....
```

**Output:** test/1.out

```
24
55 E9 1C BD 55 7A
9, 1
9, 7
3, 7
3, 4
4, 4
4, 1
```

3016.86 ms

**Tangkapan Layar**

```

C:\Users\Azmi\projects\Tucil1_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucil1_13522109\test\1.in
24
55 E9 1C BD 55 7A
9, 1
9, 7
3, 7
3, 4
4, 4
4, 1

3016.86 ms

Apakah ingin menyimpan solusi? (y/n) █

```

### 3. Tes matriks medium, sekuens medium, buffer size medium (Windows, clang++)

**Input:** test/2.in

```

9
7 7
BD 7A 7A E9 BD BD E9
BD 1C 55 BD BD 7A E9
E9 7A E9 1C 55 E9 55
1C BD BD BD 55 55 BD
7A 1C BD 55 7A BD 55
55 BD 7A 7A E9 1C 1C
55 55 E9 7A BD BD E9
100
BD 1C
6
E9 55
-46

```

.....  
.....  
.....

**Output:** test/2.out

5  
BD 55 1C E9  
1, 1  
1, 6  
6, 6  
6, 3

3928.93 ms

## Tangkapan Layar

```
C:\Users\Azmi\projects\Tucil1_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucil1_13522109\test\2.in
5
BD 55 1C E9
1, 1
1, 6
6, 6
6, 3

3928.93 ms

Apakah ingin menyimpan solusi? (y/n) █
```

4. Tes matriks medium, sekuens medium, buffer size medium, tidak memiliki solusi (Windows, clang++)

**Input:** test/3.in

```
9
6 6
7A 55 E9 1C BD 55
BD 1C BD 1C E9 BD
55 1C 1C 1C 55 55
1C 1C 55 7A BD 55
E9 1C 1C 1C BD E9
55 E9 7A 55 E9 E9
200
55 55 7A
8
BD E9 E9
-9
.....
.....
.....
```

**Output:** test/3.out

Tidak ditemukan solusi yang menghasilkan total hadiah yang positif.  
1727.25 ms

### Tangkapan Layar

```
C:\Users\Azmi\projects\Tucil1_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucil1_13522109\test\3.in
Tidak ditemukan solusi yang menghasilkan total hadiah yang positif.
1727.25 ms

Apakah ingin menyimpan solusi? (y/n) █
```



## 5. Tes row banyak, buffer medium-high (Windows, clang++)

**Input:** test/4.in

```
11
2 20
BD E9
1C 1C
BD 55
55 55
55 7A
7A 1C
7A 55
7A 55
E9 BD
E9 55
7A BD
1C 1C
55 55
55 1C
BD 55
1C 1C
BD E9
7A 1C
E9 BD
55 BD
1
BD 55 BD 1C 1C E9 BD 55 BD 1C 1C
10
```

**Output:** test/4.out

```
10
BD 55 BD 1C 1C E9 BD 55 BD 1C 1C
1, 1
1, 20
2, 20
2, 2
1, 2
1, 9
2, 9
2, 3
1, 3
1, 12
2, 12
```

96.673 ms

## Tangkapan Layar

```
C:\Users\Azmi\projects\Tucil1_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucil1_13522109\test\4.in
10
BD 55 BD 1C 1C E9 BD 55 BD 1C 1C
1, 1
1, 20
2, 20
2, 2
1, 2
1, 9
2, 9
2, 3
1, 3
1, 12
2, 12

96.673 ms

Apakah ingin menyimpan solusi? (y/n) █
```

## 6. Tes row banyak, buffer banyak (Windows, clang++)

**Input:** test/5.in

```
19
2 10
BD 1C
BD 7A
BD 55
BD E9
1C 7A
1C 55
1C E9
7A 55
7A E9
55 E9
1
1C E9 55 BD 7A E9 7A BD 55 55 7A BD E9 E9 1C 1C 7A 55 1C
10
```

**Output:** test/5.out

```
10
1C E9 55 BD 7A E9 7A BD 55 55 7A BD E9 E9 1C 1C 7A 55 1C
2, 1
2, 10
1, 10
1, 2
2, 2
2, 9
1, 9
1, 3
2, 3
2, 8
1, 8
1, 4
2, 4
2, 7
1, 7
1, 5
2, 5
2, 6
1, 6
```

77.2343 ms

**Tangkapan Layar**

```

C:\Users\Azmi\projects\Tucill1_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucill1_13522109\test\5.in
10
1C E9 55 BD 7A E9 7A BD 55 55 7A BD E9 E9 1C 1C 7A 55 1C
2, 1
2, 10
1, 10
1, 2
2, 2
2, 9
1, 9
1, 3
2, 3
2, 8
1, 8
1, 4
2, 4
2, 7
1, 7
1, 5
2, 5
2, 6
1, 6

77.2343 ms

Apakah ingin menyimpan solusi? (y/n) 

```

7. Tes matriks medium, sekuens medium, buffer size medium, tidak memiliki solusi (Windows, clang++)

**Input:** test/6.in

```

20
10 2
BD 1C 7A 55 E9 BD 1C 7A 55 E9
E9 BD 1C 7A 55 E9 7A 55 E9 BD
1
E9 BD E9 55 7A 55 7A 1C BD E9 55 E9 55 7A 1C 7A 1C BD E9 BD
10

```

## Output: test/6.out

```
10
E9 BD E9 55 7A 55 7A 1C BD E9 55 E9 55 7A 1C 7A 1C BD E9 BD
10, 1
10, 2
9, 2
9, 1
8, 1
8, 2
7, 2
7, 1
1, 1
1, 2
5, 2
5, 1
4, 1
4, 2
3, 2
3, 1
2, 1
2, 2
6, 2
6, 1
```

239.277 ms

## Tangkapan Layar

```
C:\Users\Azmi\projects\Tucill_13522109>bin\windows_10.exe
Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Bruce Force.

Pilih metode input:
1. Input dengan text file.
2. Input dihasilkan secara otomatis.

:> 1
Format text file:

buffer_size
matrix_width matrix_height
matrix (jumlah row sebanyak matrix_height dan jumlah kolom sebanyak matrix_width)
number_of_sequences
sequences_1
sequences_1_reward
sequences_2
sequences_2_reward
...
sequences_n
sequences_n_reward

Masukkan alamat lengkap file: C:\Users\Azmi\projects\Tucill_13522109\test\6.in
10
E9 BD E9 55 7A 55 7A 1C BD E9 55 E9 55 7A 1C 7A 1C BD E9 BD
10, 1
10, 2
9, 2
9, 1
8, 1
8, 2
7, 2
7, 1
1, 1
1, 2
5, 2
5, 1
4, 1
4, 2
3, 2
3, 1
2, 1
2, 2
6, 2
6, 1

239.277 ms

Apakah ingin menyimpan solusi? (y/n) █
```

#### 4. Lampiran dan Lain-lain

Link github: [https://github.com/RealAzzmi/Tucil1\\_13522109](https://github.com/RealAzzmi/Tucil1_13522109)

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil dijalankan	✓	
3. Program dapat membaca masukan berkas .txt	✓	
4. Program dapat menghasilkan masukan secara acak	✓	
5. Solusi yang diberikan program optimal	✓	
6. Program dapat menyimpan solusi dalam berkas .txt	✓	
7. Program memiliki GUI		✓