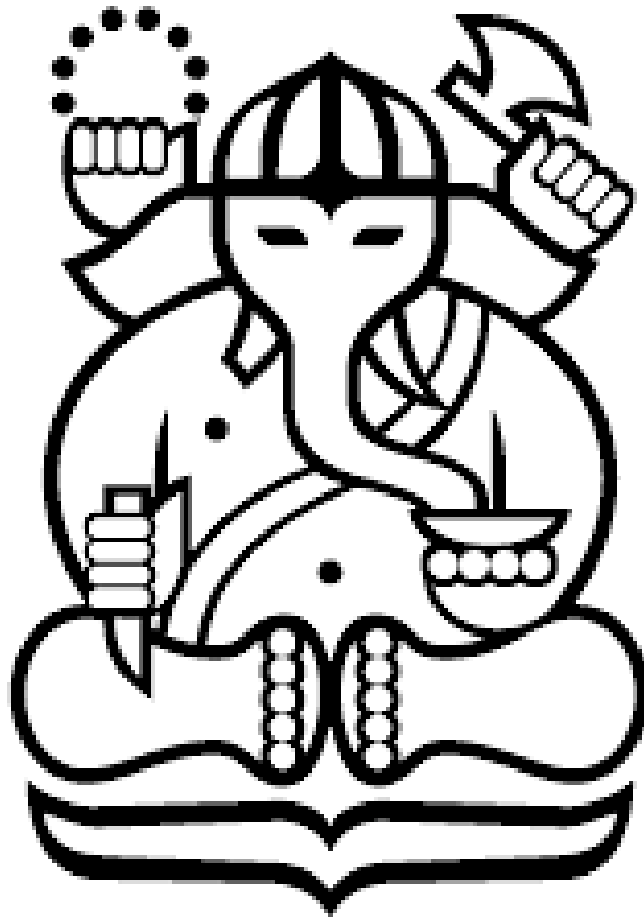


Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma
Brute Force



Disusun oleh:

Dzaky Satrio Nugroho

13522059

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024

Daftar Isi

Daftar Isi	2
Bab 1: Deskripsi Masalah	4
Bab 2: Landasan Teori	5
2.1. Brute Force	5
2.2. Algoritma	5
Bab 3: Source Program	6
Bab 4: Implementasi dan Uji Coba	13
Daftar Pustaka	17
Link Repository	17

Bab 1: Deskripsi Masalah

Cyberpunk 2077 Breach Protocol adalah *minigame* meretas pada permainan video *Cyberpunk 2077*. *Minigame* ini merupakan simulasi peretasan jaringan local dari *ICE (Intrusion Countermeasures Electronics)* pada permainan *Cyberpunk 2077*. Komponen pada permainan ini antara lain adalah:

1. Token – terdiri dari dua karakter alfanumerik seperti E9, BD, dan 55.
2. Matriks – terdiri atas token-token yang akan dipilih untuk menyusun urutan kode.
3. Sekuens – sebuah rangkaian token (dua atau lebih) yang harus dicocokkan.
4. Buffer – jumlah maksimal token yang dapat disusun secara sekuensial.

Aturan permainan Breach Protocol antara lain:

1. Pemain bergerak dengan pola vertikal, horizontal, vertikal (bergantian) hingga semua sekuens berhasil dicocokkan atau buffer penuh.
2. Pemain memulai dengan memilih satu token pada posisi baris paling atas dari matriks.
3. Sekuens dicocokkan pada token-token yang berada di buffer.
4. Satu token pada buffer dapat digunakan pada lebih dari satu sekuens.
5. Setiap sekuens memiliki bobot hadiah atau reward yang variatif.
Sekuens memiliki panjang minimal berupa dua token.

Bab 2: Landasan Teori

2.1. *Brute Force*

Brute force adalah cara yang lempang untuk menyelesaikan sebuah permasalahan.

2.2. Algoritma

Algoritma yang digunakan untuk menemukan solusi dari permainan Cyberpunk 2077 Breach Protocol adalah dengan menggunakan pendekatan brute force dengan langkah langkah sebagai berikut:

1. Inisialisasi sebuah variabel untuk menyimpan poin maksimal dengan 0.
2. Inisialisasi array kosong untuk menyimpan jalur yang dilewati.
3. Pilih 1 token pada baris pertama, masukkan pada array jalur.
4. Hitung nilai sekarang, jika lebih tinggi maka ganti nilai variabel dengan nilai sekarang, dan jalur sekarang pada array jawaban, jika nilai sama dan jalur lebih pendek maka ganti jalur dengan yang lebih pendek. Panggil fungsi untuk semua token di kolom yang sama yang belum ada pada array jalur, tambahkan masing masing token pada array jalur.
5. Hitung nilai sekarang, jika lebih tinggi maka ganti nilai variabel dengan nilai sekarang, dan jalur sekarang pada array jawaban, jika nilai sama dan jalur lebih pendek maka ganti jalur dengan yang lebih pendek. Panggil fungsi untuk semua token di baris yang sama yang belum ada pada array jalur, tambahkan masing masing token pada array jalur.
6. Ulangi langkah 4 dan 5 hingga ukuran array jalur sebesar buffer atau tidak dapat memilih token lagi.

Bab 3: Source Program

```
long long int tokenSize, bufferSize, matrixWidth, matrixHeight, numberOfSequence, maxSequence, pointMax = 0;
vector<int> point = {}, pathMax = {};
vector<string> sequence = {};
vector<vector<string>> matrix;
```

Variabel Global

```
string RemoveSpace(string str){
    int size = str.length();
    for(int i=0;i<=size;i++){
        for(int j=0;j<=i;j++){
            if(str[j] == ' '){
                str.erase(str.begin() + j);
            }
        }
    }
    return str;
}
```

Fungsi menghapus karakter spasi di string

```

void HitungPoin(vector<int> path){
    long long int i, points = 0;
    string str = "";

    // Mengubah path menjadi string
    for(i=0;i<path.size();i++){
        str.append(matrix[(path[i])/matrixWidth][(path[i])%matrixWidth]);
    }

    // Menambahkan poin untuk tiap sequence yang muncul
    for(i=0;i<sequence.size();i++){
        if(str.find(sequence[i]) != -1){
            points += point[i];
        }
    }

    // Mengganti nilai pointMax dan path jika nilai poin lebih besar dari poin sekarang
    if(points > pointMax){
        pointMax = points;
        pathMax = path;
    }

    // Untuk mendapatkan jawaban dengan banyak token minimal
    if(points == pointMax && path.size() < pathMax.size()){
        pointMax = points;
        pathMax = path;
    }
}

```

Fungsi untuk menghitung poin

```

void iterate(vector<int> path){
    long long int mods,i;
    // Jika awal iterasi
    if(path.size() == 0){
        for(i=0;i<matrixWidth;i++){
            path = {};
            path.push_back(i);
            iterate(path);
        }
    } else if(path.size() == bufferSize){ // Jika Buffer sudah penuh
        HitungPoin(path);
    } else if(path.size() % 2){ // Jika gerakan sebelumnya horizontal
        mods = path.back() % matrixWidth;
        for(i=0;i<matrixHeight;i++){
            vector<int> newPath;
            if(find(path.begin(),path.end(),(matrixWidth*i) + mods) == path.end()){
                newPath = path;
                newPath.push_back((matrixWidth*i) + mods);
                iterate(newPath);
            }
        }
        HitungPoin(path);
    } else { // Jika gerakan sebelumnya vertikal
        mods = (path.back() / matrixWidth) * matrixWidth;
        for(i=mods;i<mods+matrixWidth;i++){
            vector<int> newPath;
            if(find(path.begin(),path.end(),i) == path.end()){
                newPath = path;
                newPath.push_back(i);
                iterate(newPath);
            }
        }
        HitungPoin(path);
    }
}

```

Fungsi Brute Force utama


```

// Membaca Sekeuns
getline(file,line);
numberOfSequence = stoi(line);
for(long long int i=0;i<numberOfSequence;i++){
    getline(file,line);
    sequence.push_back(RemoveSpace(line));
    getline(file,line);
    point.push_back(stoi(line));
}
file.close();
} else {
    cout << "\nFile txt tidak ditemukan pada folder test, pastikan penulisan sudah benar!\n";
}
}

```

Input dari file

```

cout << "Masukkan jumlah token unik : ";
cin >> tokenSize;
string token[tokenSize];

cout << "Masukkan token unik(dipisahkan dengan spasi): ";
cin.ignore();
getline(cin,userInput);
for(long long int i=0;i<tokenSize;i++){
    token[i] = userInput.substr(3*i,2);
}

cout << "Masukkan ukuran buffer : ";
cin >> bufferSize;
srand(time(0));

cout << "Masukkan ukuran matrix(Lebar Panjang) : ";
cin.ignore();
getline(cin,userInput);
int temp = userInput.length();
size_t found = userInput.find(" ");
matrixWidth = stoi(userInput.substr(0,found));
matrixHeight = stoi(userInput.substr(found+1,temp-found));
for(long long int i=0;i<matrixHeight;i++){
    vector<string> temps = {};
    for(long long int j=0;j<matrixWidth;j++){
        temps.push_back(token[rand() % tokenSize]);
    }
    matrix.push_back(temps);
}
}

```

Input dari CLI

```

cout << "Masukkan jumlah sekuens : ";
cin >> numberOfSequence;

cout << "Masukkan ukuran maksimal sekuens : ";
cin >> maxSequence;
for(long long int i=0;i<numberOfSequence;i++){
    string temps = "";
    long long int gacha = (rand() % (maxSequence - 2)) + 2;
    for(long long int j=0;j<gacha;j++){
        temps += token[rand() % tokenSize];
    }
    sequence.push_back(temps);
    point.push_back((rand() % 200) - 100);
}

cout << "Matriks permainan :\n";
for(long long int i=0;i<matrixHeight;i++){
    for(long long int j=0;j<matrixWidth;j++){
        cout << matrix[i][j] << " ";
    }
    cout << endl;
}

cout << "Sekuens permainan :\n";
for(long long int i=0;i<numberOfSequence;i++){
    for(long long int j=0;j<sequence[i].length();j++){
        cout << sequence[i][j];
        if(j%2){
            cout << " ";
        }
    }
    cout << endl << point[i] << endl;
}

```

Input dari CLI dan menghasilkan matriks dan sekuens random

```

// Memulai Bruteforce
auto start = chrono::high_resolution_clock::now();
iterate({});
auto end = chrono::high_resolution_clock::now();
auto duration = chrono::duration_cast<std::chrono::milliseconds>(end - start);

// Output
cout << "Reward : " << pointMax;
cout << "\nPath : ";
for(int i=0;i<pathMax.size();i++){
    cout << matrix[pathMax[i]/matrixWidth][pathMax[i]%matrixWidth] << " ";
}
cout << "\nCoordinate :\n";
for(int i=0;i<pathMax.size();i++){
    cout << pathMax[i]%matrixWidth + 1 << ", " << pathMax[i]/matrixWidth + 1 << endl;
}
cout << "\nTime taken : " << duration.count() << " ms" << endl;

```

Memulai fungsi brute force dan menghitung waktu serta output di CLI

```

cout << "Apakah ingin menyimpan solusi? (y/n)\n";
cin >> userInput;
while(userInput.length() != 1 || !(userInput[0] == 89 || userInput[0] == 121 || userInput[0] == 78 || userInput[0] == 110)){
    cout << "\nMasukkan tidak valid.\nApakah ingin menyimpan solusi? (y/n)\n";
    cin >> userInput;
}

if(userInput == "y" || userInput == "Y"){
    cout << "Masukkan nama file untuk disimpan : ";
    cin.ignore();
    getline(cin,userInput);
    userInput.insert(0,"../test/");

    ofstream outputFile(userInput);

    if (!outputFile.is_open()) {
        cerr << "Error opening file for writing!" << endl;
        return 1;
    }

    // Menuliskan data ke file external
    outputFile << "Reward : " << pointMax;
    outputFile << "\nPath : ";
    for(int i=0;i<pathMax.size();i++){
        outputFile << matrix[pathMax[i]/matrixWidth][pathMax[i]%matrixWidth] << " ";
    }
    outputFile << "\nCoordinate :\n";
    for(int i=0;i<pathMax.size();i++){
        outputFile << pathMax[i]%matrixWidth + 1 << ", " << pathMax[i]/matrixWidth + 1 << endl;
    }
    outputFile << "\nTime taken : " << duration.count() << " ms" << endl;

    outputFile.close();

    cout << "File telah berhasil disimpan!" << endl;
}

```

Menyimpan output dalam file

Bab 4: Implementasi dan Uji Coba

Test case 1

```
test > ≡ tc1.txt
1    7
2    6 6
3    7A 55 E9 E9 1C 55
4    55 7A 1C 7A E9 55
5    55 1C 1C 55 E9 BD
6    BD 1C 7A 1C 55 BD
7    BD 55 BD 7A 1C 1C
8    1C 55 55 7A 55 7A
9    3
10   BD E9 1C
11   15
12   BD 7A BD
13   20
14   BD 1C BD 55
15   30
```

```
Selamat datang di
-----
Bread Protocol
-----
INPUT :
1. txt
2. Random
Silahkan pilih metode input : 1

Metode input txt telah dipilih.
Masukkan nama file txt yang ingin dipakai (pastikan sudah ada dalam folder test): tc1.txt
Reward : 50
Path : 7A BD 7A BD 1C BD 55
Coordinate :
1, 1
1, 4
3, 4
3, 5
6, 5
6, 3
1, 3

Time taken : 242 ms
Apakah ingin menyimpan solusi? (y/n)
n
Terima kasih telah menggunakan program ini!
```

Test case 2

```
test > ≡ tc2.txt
1 7
2 10 8
3 BD 1C 55 55 F3 E9 55 F3 E9 8G
4 55 E9 1C 7A 8G 8G 55 E9 55 55
5 F3 55 F3 8G E9 1C 8G E9 F3 1C
6 E9 7A 1C F3 55 1C E9 7A 8G F3
7 1C 55 F3 7A 8G 8G F3 8G 55 8G
8 1C 8G BD E9 E9 BD 8G 7A 55 E9
9 1C 55 F3 E9 7A 1C BD 7A 7A F3
10 F3 1C 55 BD 55 1C 8G 55 1C 1C
11 4
12 F3 E9 E9
13 32
14 8G BD E9
15 34
16 BD BD F3 E9
17 24
18 E9 8G F3
19 20
```

```
Selamat datang di
-----
Reach Protocol
-----
INPUT :
1. txt
2. Random
Silahkan pilih metode input : 1

Metode input txt telah dipilih.
Masukkan nama file txt yang ingin dipakai (pastikan sudah ada dalam folder test): tc2.txt
Reward : 66
Path : F3 E9 E9 8G BD E9
Coordinate :
5, 1
5, 6
10, 6
10, 1
1, 1
1, 4

Time taken : 6187 ms
Apakah ingin menyimpan solusi? (y/n)
n
Terima kasih telah menggunakan program ini!
```

Test case 3

```
test > tc3.txt
1 7
2 12 9
3 U8 55 7A 2L U8 U8 2L 55 2L E9 E9 55
4 U8 55 BD 2L 1C 2L 55 BD 2L BD 55 BD
5 U8 55 BD BD 7A 7A U8 U8 BD 7A U8 1C
6 1C 1C U8 2L BD E9 U8 U8 7A 7A 7A E9
7 BD 55 2L 2L BD BD 2L 1C 1C 7A 2L 1C
8 55 BD 55 1C 2L BD 7A U8 1C 2L BD E9
9 BD BD 55 55 1C 2L 55 BD 7A E9 55 55
10 BD 55 U8 55 2L 1C 2L 55 1C BD 7A 2L
11 BD 7A 2L BD 7A 1C 7A 55 7A 2L U8 1C
12 5
13 7A 7A 55
14 14
15 55 55
16 21
17 E9 55 2L
18 31
19 2L 55 1C BD 2L 7A
20 46
21 1C 2L E9
22 9
```

```
Selamat datang di
-----
Breach Protocol
-----
INPUT :
1. txt
2. Random
Silahkan pilih metode input : 1

Metode input txt telah dipilih.
Masukkan nama file txt yang ingin dipakai (pastikan sudah ada dalam folder test): tc3.txt
Reward : 66
Path : E9 55 2L 7A 7A 55 55
Coordinate :
11, 1
11, 2
9, 2
9, 4
11, 4
11, 7
3, 7

Time taken : 21192 ms
Apakah ingin menyimpan solusi? (y/n)
y
Masukkan nama file untuk disimpan : output.txt
File telah berhasil disimpan!
```

```

test > ≡ output.txt
1   Reward : 66
2   Path : E9 55 2L 7A 7A 55 55
3   Coordinate :
4   11, 1
5   11, 2
6   9, 2
7   9, 4
8   11, 4
9   11, 7
10  3, 7
11
12  Time taken : 21192 ms

```

Test case 4

```

-----
INPUT :
1. txt
2. Random
Silahkan pilih metode input : 2
Masukkan jumlah token unik : 5
Masukkan token unik(dipisahkan dengan spasi): AA BB CC DD EE
Masukkan ukuran buffer : 6
Masukkan ukuran matrix(Lebar Panjang) : 5 5
Masukkan jumlah sekuens : 4
Masukkan ukuran maksimal sekuens : 4
Matriks permainan :
BB AA DD BB DD
AA CC CC AA CC
CC AA AA EE DD
CC BB DD CC BB
BB CC BB CC DD
Sekuens permainan :
AA BB
17
CC CC AA
15
CC AA
0
BB EE
88
Reward : 120
Path : DD CC CC AA BB EE
Coordinate :
3, 1
3, 2
2, 2
2, 1
4, 1
4, 3

Time taken : 21 ms
Apakah ingin menyimpan solusi? (y/n)
n
Terima kasih telah menggunakan program ini!

```

[illegible]


```

-----
INPUT :
1. txt
2. Random
Silahkan pilih metode input : 2
Masukkan jumlah token unik : 6
Masukkan token unik(dipisahkan dengan spasi): Z1 Y2 X3 C4 6V 7J
Masukkan ukuran buffer : 4
Masukkan ukuran matrix(Lebar Panjang) : 6 7
Masukkan jumlah sekuens : 5
Masukkan ukuran maksimal sekuens : 3
Matriks permainan :
Y2 C4 Y2 C4 Z1 6V
X3 Y2 X3 Z1 Z1 6V
X3 6V 6V C4 X3 7J
7J Z1 7J Z1 7J C4
6V 7J C4 Z1 C4 C4
6V 6V Z1 6V 7J X3
C4 7J C4 6V 6V 6V
Sekuens permainan :
X3 Z1
86
7J C4
33
7J C4
62
C4 6V
-5
Z1 7J
12
Reward : 107
Path : Z1 7J C4
Coordinate :
5, 1
5, 4
6, 4

Time taken : 4 ms
Apakah ingin menyimpan solusi? (y/n)
y
Masukkan nama file untuk disimpan : output2.txt
File telah berhasil disimpan!

```

```

test > ≡ output2.txt
1   Reward : 107
2   Path : Z1 7J C4
3   Coordinate :
4   5, 1
5   5, 4
6   6, 4
7
8   Time taken : 4 ms

```

Daftar Pustaka

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		✓

Link Repository

https://github.com/Kizaaaa/Tucil1_13522059