

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma
Penyelesaian *Cyberpunk 2077 Breach Protocol* dengan Algoritma Brute
Force

Semester II Tahun 2023/2024



disusun oleh:
Nicholas Reymond Sihite (13522144)

INSTITUT TEKNOLOGI BANDUNG
2024

BAB I

Deskripsi Masalah

Cyberpunk 2077 adalah sebuah permainan video dengan genre *role-playing* yang dirilis pada tahun 2020. Terdapat sebuah permainan kecil (*minigame*) yang dapat dimainkan pada *Cyberpunk 2077* yang bernama *Breach Protocol*. Permainan kecil ini dimainkan dengan cara memilih salah satu kotak dari matriks token yang disediakan, lalu mencari sekuens dengan pola vertikal-horizontal secara bergantian sehingga didapat sekuens paling optimal (menghasilkan *reward* terbesar).

Tidak sedikit orang yang kesulitan dalam mencari solusi paling optimal dalam *Breach Protocol*. Oleh sebab itu, dibutuhkan sebuah program untuk mencari solusi optimal sehingga pemain dapat mendapatkan *reward* maksimal pula. Pada program yang telah dibuat ini, algoritma yang digunakan untuk menyelesaikan masalah adalah algoritma *brute force* yang diimplementasikan dengan bahasa pemrograman C.

BAB II

Analisis Algoritma

Terdapat 5 *abstract data type* yang digunakan pada program ini, yaitu *dynamic array of integer*, *dynamic array of string*, *dynamic matrix of string*, *point*, dan *dynamic array of point*. Matriks token dan daftar sekuens disimpan sebagai *dynamic matrix of string*, Panjang sekuens dan *reward*-nya disimpan sebagai *dynamic array of integer*, dan rute yang diambil disimpan sebagai *dynamic array of point*.

Pada bagian awal, program akan menanyakan ingin menerima *input* dari *file* atau CLI. Jika dari *file*, pengguna harus memasukkan nama *file* yang ingin dibaca (*file* harus berada pada folder “test”). Jika pengguna ingin memasukkan *input* dari CLI, akan ada beberapa masukan yang akan diterima program. Semua *input* akan divalidasi oleh program sehingga jika ada kesalahan: kesalahan pada *file input*, program akan mengeluarkan pesan gagal dan keluar otomatis dalam 5 detik; kesalahan pada CLI *input*, program akan meminta masukan kembali. Pada pembentukan sekuens secara *random*, program juga melakukan validasi dengan mengecek apakah sekuens baru sudah ada sebelumnya atau tidak (jika ya, sekuens *random* baru akan dibuat). *Reward* tiap sekuens dihasilkan secara acak dengan kelipatan 5 pada rentang -50 sampai 50 dengan kemungkinan menghasilkan *reward* positif 80% dan *reward* negatif 20%.

Algoritma *brute force* adalah satu jenis algoritma yang digunakan dengan cara memeriksa seluruh kemungkinan yang ada. Pada konteks masalah *Breach Protocol*, di awal permainan pemain harus memilih salah satu dari token yang ada pada baris pertama matriks. Oleh sebab itu, dibuat sebuah perulangan (*loop*) yang mengulang perhitungan *reward* dari sudut kiri atas ke sudut kanan atas matriks. Setelah itu, akan dilakukan *traversing* dengan seluruh kemungkinan rute yang ada secara rekursif dengan sebuah prosedur bernama *recursiveTraverse*.

Basis dari rekursi yang telah dibuat adalah kasus ketika *buffer* penampung sekuens sudah penuh. Untuk rekurensnya sendiri, ada 2 kemungkinan, yaitu pergerakan vertikal dan horizontal. Sesuai namanya, pergerakan vertikal hanya memungkinkan langkah selanjutnya yang diambil berada pada kolom yang sama, sementara pergerakan horizontal pada baris yang sama.

Setiap kali ada penambahan titik, *reward* yang mungkin dihasilkan akan diperiksa dengan sebuah fungsi buatan yang bernama *addReward*. Jika *reward* yang dihasilkan melebihi *reward* maksimal sebelumnya, rute dan besar *reward* akan disimpan. Setelah semua kemungkinan rute telah dijalani, program akan menampilkan *reward* terbesar yang mungkin, sekuens yang telah diambil, (salah satu) rute yang menghasilkan *reward* tersebut, beserta waktu yang dibutuhkan untuk mengeksekusi bagian algoritma *brute force* pada program.

Setelah solusi optimal ditemukan (bisa jadi tidak ada solusi), program akan menanyakan apakah pengguna ingin menyimpan hasilnya atau tidak. Jika tidak, program akan langsung tertutup. Jika ya, program akan meminta *input* berupa nama *file* yang diinginkan. Penyimpanan *file* .txt akan dilakukan pada folder “test”. Program juga akan langsung tertutup ketika penyimpanan file selesai dilakukan.

Terdapat 1 file utama dan 3 file tambahan untuk mendukung keberjalanannya program, yaitu “main.c”, “boolean.h”, “cyberpunkhack.h”, dan “cyberpunkhack.c”. Berikut merupakan *source code* file *main.c* (untuk file lain dapat dilihat pada *repository* yang berada pada bagian referensi).

```

        } else {
            printf("Masukan tidak valid!\n");
        }
    }

if (option == 1){ // Jika input dari file
    printf("Masukan Anda adalah dari file\n");

    // input nama file
    valid = false;
    while (!valid){
        printf("Silakan masukkan nama file yang ingin Anda buka (maksimal 200
karakter, jangan lupa tambahkan .txt pada akhir nama file)\n");
        printf(">> "); scanf("%199s", fileName);

        sprintf(filePath, "../test/%s", fileName);
        filePointer = fopen(filePath, "r");
        if (filePointer == NULL){
            printf("File tidak ditemukan! Silakan masukkan nama file yang
benar!\n");
        } else {
            valid = true;
        }
    }

    // membaca input dari file
    fscanf(filePointer, "%d", &bufferSize);
    if (bufferSize <= 0){
        printf("Ukuran buffer tidak valid.\n");
        printf("Program akan dihentikan dalam 5 detik, silakan perbaiki file
input.\n");
        sleep(5);
        return 0;
    }

    fscanf(filePointer, "%d %d", &mtxHeight, &mtxWidth);
    if ((mtxHeight <= 0) || (mtxWidth <= 0)){
        printf("Ukuran matrix tidak valid.\n");
        printf("Program akan dihentikan dalam 5 detik, silakan perbaiki file
input.\n");
        sleep(5);
        return 0;
    }

    createStringMatrix(&tokenMatrix, mtxHeight, mtxWidth);
    for (i = 0; i < mtxHeight; i++){
        for (j = 0; j < mtxWidth; j++){
            fscanf(filePointer, "%s", Word);
            if (strlen(Word) != 2){
                printf("Panjang token harus 2.\n");
                printf("Kesalahan ada pada baris %d kolom %d -> %s\n", i + 1,
j + 1, Word);
                printf("Program akan dihentikan dalam 5 detik, silakan perbaiki
file input.\n");
                sleep(5);
                return 0;
            }
            tokenMatrix.buffer[i].buffer[j] = strdup(Word);
        }
    }

    fscanf(filePointer, "%d", &cntSeq);
    if (cntSeq <= 0){
        printf("Banyaknya sequence tidak valid.\n");
        printf("Program akan dihentikan dalam 5 detik, silakan perbaiki file
input.\n");
        sleep(5);
        return 0;
    }
}

```

```

}

createStringMatrix(&sequences, cntSeq, mtxWidth);
createIntArray(&seqSize, cntSeq);
createIntArray(&seqReward, cntSeq);
for (i = 0; i < cntSeq; i++) {
    do {
        fgets(line, sizeof(line), filePointer);
    } while (strcmp(line, "\n") == 0);

    token = strtok(line, " ");
    j = 0;
    while (token != NULL) {
        nlIdx = strcspn(token, "\n");
        if (token[nlIdx] == '\n'){
            token[nlIdx] = '\0';
        }
        if (strlen(token) != 2){
            printf("Panjang token harus 2.\n");
            printf("Kesalahan ada pada baris %d -> %s\n", i + 1, token);
            printf("Program akan dihentikan dalam 5 detik, silakan perbaiki
file input.\n");
            sleep(5);
            return 0;
        }
        sequences.buffer[i].buffer[j] = strdup(token);
        j++;
        token = strtok(NULL, " ");
    }
    seqSize.buffer[i] = j;
    if (fscanf(filePointer, "%d", &seqReward.buffer[i]) == 1){
        fscanf(filePointer, "%d", &seqReward.buffer[i]);
    } else {
        printf("Reward sequence ke-%d tidak valid.\n", i + 1);
        printf("Program akan dihentikan dalam 5 detik, silakan perbaiki
file input.\n");
        sleep(5);
        return 0;
    }
}

fclose(filePointer);

printf("\nBerikut merupakan matrix sequence dari file\n");
displayMatrix(tokenMatrix);

printf("\nBerikut merupakan sequence unik dan reward random\n");
displaySeqAndReward(sequences, seqSize, seqReward);
printf("\n");

} else { // Jika input dari CLI (setelah validasi, hanya mungkin option == 2)
    printf("Masukan Anda adalah dari CLI\n");

    valid = false;
    while (!valid){ // validasi banyaknya token unik (harus lebih dari 0)
        printf("Masukkan banyaknya token unik yang ingin dijadikan matrix
sequence\n");
        printf(">> "); scanf("%d", &cntToken);
        if (cntToken > 0){
            valid = true;
        } else {
            printf("Banyaknya token unik harus lebih dari 0!\n");
        }
    }

    // membuat array untuk menyimpan token unik
    createStringArray(&seqArray, cntToken);
}

```

```

// menerima input dari user
printf("Masukkan token unik yang ingin dijadikan matrix sequence\n");
for (i = 0; i < cntToken; i++){
    scanf("%s", Word);
    if (isInStrArray(seqArray, Word)){
        do {
            printf("Token harus unik! Silakan masukkan ulang token!\n");
            scanf("%s", Word);
        } while (isInStrArray(seqArray, Word));
    }
    if (strlen(Word) != 2){
        do {
            printf("Panjang token harus 2! Silakan masukkan ulang
token!\n");
            scanf("%s", Word);
        } while (strlen(Word) != 2);
    }
    insertLastStrArr(&seqArray, Word);
}

valid = false;
while (!valid){
    printf("Masukkan ukuran buffer yang diinginkan\n");
    printf(">> "); scanf("%d", &bufferSize);
    if (bufferSize > 0){
        valid = true;
    } else {
        printf("Ukuran buffer harus lebih dari 0!\n");
    }
}

valid = false;
while (!valid){
    printf("Masukkan ukuran matrix sequence yang diinginkan\n");
    printf(">> "); scanf("%d %d", &mtxHeight, &mtxWidth);
    if ((mtxHeight > 0) && (mtxWidth > 0)){
        valid = true;
    } else {
        printf("Ukuran matrix harus lebih dari 0!\n");
    }
}

valid = false;
while (!valid){
    printf("Masukkan banyaknya sequence yang diinginkan\n");
    printf(">> "); scanf("%d", &cntSeq);
    if (cntSeq > 0){
        valid = true;
    } else {
        printf("Banyaknya sequence harus lebih dari 0!\n");
    }
}

valid = false;
while (!valid){
    printf("Masukkan ukuran maksimal sequence yang diinginkan\n");
    printf(">> "); scanf("%d", &maxSeqSize);
    if (maxSeqSize > 0){
        valid = true;
    } else {
        printf("Ukuran maksimal sequence harus lebih dari 0!\n");
    }
}

// membuat matrix random
createStringMatrix(&tokenMatrix, mtxHeight, mtxWidth);
rand(time(NULL));
for (i = 0; i < mtxHeight; i++) {

```

```

        for (j = 0; j < mtxWidth; j++) {
            randomInt = rand() % cntToken;
            tokenMatrix.buffer[i].buffer[j] = seqArray.buffer[randomInt];
        }
    }
    printf("\nBerikut merupakan matrix sequence random\n");
    displayMatrix(tokenMatrix);

    // membuat random sequence
    createStringMatrix(&sequences, cntSeq, maxSeqSize);
    createIntArray(&seqSize, cntSeq);
    createIntArray(&seqReward, cntSeq);
    for (i = 0; i < cntSeq; i++){
        // panjang sequence random
        tempSeqLength = (rand() % maxSeqSize) + 1;
        seqSize.buffer[i] = tempSeqLength;

        // sequence random
        for (j = 0; j < tempSeqLength; j++){
            randomInt = rand() % cntToken;
            sequences.buffer[i].buffer[j] = seqArray.buffer[randomInt];
        }
        while (isSequenceDup(sequences, seqSize, sequences.buffer[i],
tempSeqLength, i)){ // loop untuk memeriksa sequence duplikat
            for (j = 0; j < tempSeqLength; j++){
                randomInt = rand() % cntToken;
                sequences.buffer[i].buffer[j] = seqArray.buffer[randomInt];
            }
        }
    }

    // hadiah sequence random berupa integer kelipatan 5 pada range -50
    sampai 50
    randomInt = ((rand() % 10) + 1) * 5;
    reward = randomInt;
    randomInt = rand() % 5;
    if (randomInt == 1){ // jika randomInt == 1, maka reward sequence
negatif (peluang 1/5 untuk mendapatkan reward negatif)
        reward = -reward;
    }
    seqReward.buffer[i] = reward;
}
printf("\nBerikut merupakan sequence unik dan reward random\n");
displaySeqAndReward(sequences, seqSize, seqReward);
printf("\n");
}

// time keeper
start = clock();

// algoritma brute force untuk mencari solusi optimal
currReward = 0;
maxReward = 0;
createStringArray(&bufferSolution, bufferSize);
createPointArray(&solutionCoordinate, bufferSize);
for (i = 0; i < tokenMatrix.col; i++){
    solutionCoordinate.buffer[0].X = 0;
    solutionCoordinate.buffer[0].Y = i;
    solutionCoordinate.nEff = 1;
    currReward = addReward(sequences, seqSize, seqReward, solutionCoordinate,
tokenMatrix);
    if (currReward > maxReward){
        createPointArray(&maxArr, bufferSize);
        maxReward = currReward;
        copyPointArray(solutionCoordinate, &maxArr);
    }
    recursiveTraverse(sequences, seqSize, seqReward, &solutionCoordinate,
tokenMatrix, &maxReward, &maxArr, 0, i, true);
}

```

```

// time keeper
end = clock();
interval = ((double)(end - start) / CLOCKS_PER_SEC) * 1000;

if (maxReward <= 0){
    maxReward = 0;
    printf("Tidak ada solusi!\n");
} else {
    printf("%d\n", maxReward);
    for (i = 0; i < maxArr.nEff; i++) {
        printf("%s",
tokenMatrix.buffer[maxArr.buffer[i].X].buffer[maxArr.buffer[i].Y]);
    }
    printf("\n");
    for (i = 0; i < maxArr.nEff; i++) {
        printf("%d, %d\n", maxArr.buffer[i].Y + 1, maxArr.buffer[i].X + 1);
    }
    printf("\n");
}
printf("%f ms\n", interval);

// menyimpan hasil ke dalam file
printf("Apakah Anda ingin menyimpan solusi ke dalam file .txt?\n");
printf("1. Ya\n");
printf("2. Tidak\n");

// Validasi masukan
valid = false;
while (!valid){
    printf("Silakan masukkan nomor pilihan Anda!\n");
    printf(">> "); scanf("%d", &option);
    if (option == 1 || option == 2){
        valid = true;
    } else {
        printf("Masukan tidak valid!\n");
    }
}
if (option == 1){
    printf("Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan
lupa tambahkan .txt pada akhir nama file)\n");
    printf(">> "); scanf("%199s", fileName);

    sprintf(filePath, "../test/%s", fileName);
    filePointer = fopen(filePath, "w");

    if (maxReward > 0){
        fprintf(filePointer, "%d\n", maxReward);
        for (i = 0; i < maxArr.nEff; i++){
            fprintf(filePointer, "%s", " ", maxArr.buffer[i].Y + 1,
tokenMatrix.buffer[maxArr.buffer[i].X].buffer[maxArr.buffer[i].Y]);
        }
        fprintf(filePointer, "\n");

        for (i = 0; i < maxArr.nEff; i++){
            fprintf(filePointer, "%d, %d\n", maxArr.buffer[i].Y + 1,
maxArr.buffer[i].X + 1);
        }
    } else {
        fprintf(filePointer, "Tidak ada solusi!\n");
    }

    fprintf(filePointer, "\n");
    fprintf(filePointer, "%f ms\n", interval);

    fclose(filePointer);
}

```

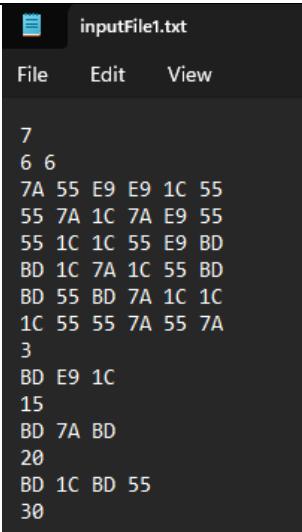
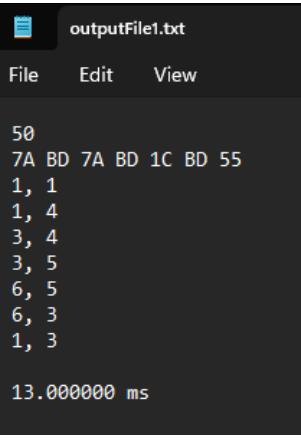
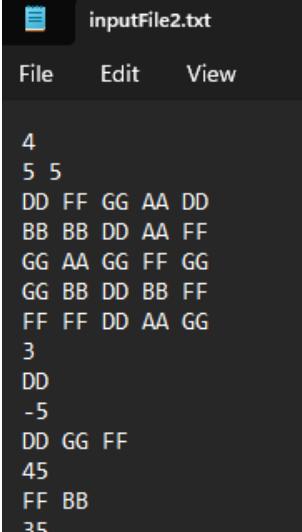
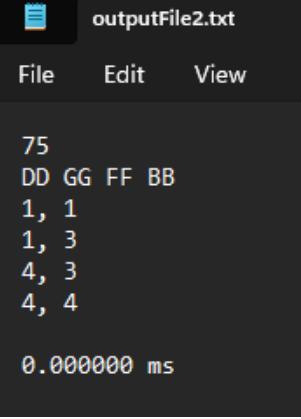
```
}
```

Berikut pula potongan kode prosedur *recursiveTraverse* yang diambil dari file “cyberpunkhack.c”

BAB III

Uji Coba Program

3.1 Masukan dari file .txt

No	Input	Output
1	 <p>inputFile1.txt File Edit View 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</p>	 <p>outputFile1.txt File Edit View 50 7A BD 7A BD 1C BD 55 1, 1 1, 4 3, 4 3, 5 6, 5 6, 3 1, 3 13.000000 ms</p>
2	 <p>inputFile2.txt File Edit View 4 5 5 DD FF GG AA DD BB BB DD AA FF GG AA GG FF GG GG BB DD BB FF FF FF DD AA GG 3 DD -5 DD GG FF 45 FF BB 35</p>	 <p>outputFile2.txt File Edit View 75 DD GG FF BB 1, 1 1, 3 4, 3 4, 4 0.000000 ms</p>

3			
4			

3.2 Masukan dari CLI secara acak

No	Input	Output
----	-------	--------

<p>1</p> <pre>Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI? 1. File 2. CLI Silakan masukkan nomor pilihan Anda! >> 2 Masukkan Anda adalah dari CLI Masukkan banyaknya token unik yang ingin dijadikan matrix sequence >> 4 Masukkan token unik yang ingin dijadikan matrix sequence AA BB CC DD Masukkan ukuran buffer yang diinginkan >> 6 Masukkan ukuran matrix sequence yang diinginkan >> 5 5 Masukkan banyaknya sequence yang diinginkan >> 4 Masukkan ukuran maksimal sequence yang diinginkan >> 3 Berikut merupakan matrix sequence random CC AA BB DD BB AA BB BB CC BB DD DD BB DD AA CC BB DD AA AA DD BB CC CC BB Berikut merupakan sequence unik dan reward random Sequence 1: CC CC BB dengan reward 35 dan panjang 3 Sequence 2: DD dengan reward 45 dan panjang 1 Sequence 3: BB dengan reward 35 dan panjang 1 Sequence 4: CC dengan reward 50 dan panjang 1 165 CC AA CC CC BB DD 1, 1 1, 2 4, 2 4, 5 2, 5 2, 3 1.000000 ms Apakah Anda ingin menyimpan solusi ke dalam file .txt? 1. Ya 2. Tidak Silakan masukkan nomor pilihan Anda! >> 1 Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file) >> testCLI1.txt</pre>	<p>testCLI1.txt</p> <p>File Edit View</p> <pre>165 CC AA CC CC BB DD 1, 1 1, 2 4, 2 4, 5 2, 5 2, 3 1.000000 ms</pre>
<p>2</p> <pre>Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI? 1. File 2. CLI Silakan masukkan nomor pilihan Anda! >> 2 Masukkan Anda adalah dari CLI Masukkan banyaknya token unik yang ingin dijadikan matrix sequence >> 4 Masukkan token unik yang ingin dijadikan matrix sequence AA BB CC DD Masukkan ukuran buffer yang diinginkan >> 7 Masukkan ukuran matrix sequence yang diinginkan >> 6 6 Masukkan banyaknya sequence yang diinginkan >> 4 Masukkan ukuran maksimal sequence yang diinginkan >> 4 Berikut merupakan matrix sequence random CC BB BB CC AA AA BB AA AA AA BB BB CC DD DD BB CC AA AA DD BB BB BB AA AA AA DD CC AA AA CC BB DD BB CC CC Berikut merupakan sequence unik dan reward random Sequence 1: DD CC CC dengan reward 20 dan panjang 3 Sequence 2: BB dengan reward 50 dan panjang 1 Sequence 3: DD dengan reward 20 dan panjang 1 Sequence 4: DD DD dengan reward 30 dan panjang 2 120 CC CC DD DD CC CC BB 1, 1 1, 3 3, 3 3, 5 4, 5 4, 1 2, 1 24.000000 ms Apakah Anda ingin menyimpan solusi ke dalam file .txt? 1. Ya 2. Tidak Silakan masukkan nomor pilihan Anda! >> 1 Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file) >> testCLI2.txt</pre>	<p>testCLI2.txt</p> <p>File Edit View</p> <pre>120 CC CC DD DD CC CC BB 1, 1 1, 3 3, 3 3, 5 4, 5 4, 1 2, 1 24.000000 ms</pre>

3

```
Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm
Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI?
1. File
2. CLI
Silakan masukkan nomor pilihan Anda!
>> 2
Masukkan Anda adalah dari CLI
Masukkan banyaknya token unik yang ingin dijadikan matrix sequence
>> 5
Masukkan token unik yang ingin dijadikan matrix sequence
A7 BF CC DO DO
Token harus unik! Silakan masukkan ulang token!
EX
Masukkan ukuran buffer yang diinginkan
>> 8
Masukkan ukuran matrix sequence yang diinginkan
>> 8
Masukkan banyaknya sequence yang diinginkan
>> 4
Masukkan ukuran maksimal sequence yang diinginkan
>> 3

Berikut merupakan matrix sequence random
CC A7 BF DO A7 CC DO A7
CC DO CC BF A7 CC CC EX
EX BF A7 CC DO A7 DO EX
A7 A7 EX EX A7 BF A7 CC
A7 A7 A7 A7 CC BF A7 EX
DO DO BF A7 A7 EX BF
DO A7 CC DO BF BF EX A7
CC BF EX A7 DO A7 CC CC

Berikut merupakan sequence unik dan reward random
Sequence 1: EX BF CC dengan reward 35 dan panjang 3
Sequence 2: DO EX dengan reward 45 dan panjang 2
Sequence 3: A7 DO dengan reward 30 dan panjang 2
Sequence 4: A7 EX dengan reward -40 dan panjang 2

110
CC CC DO A7 DO EX BF CC
1, 1
1, 2
2, 2
2, 1
4, 1
4, 4
6, 4
6, 1

1810.000000 ms
Apakah Anda ingin menyimpan solusi ke dalam file .txt?
1. Ya
2. Tidak
Silakan masukkan nomor pilihan Anda!
>> 1
Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file)
>> testCLI3.txt
```

testCLI3.txt

File Edit View

```
110
CC CC DO A7 DO EX BF CC
1, 1
1, 2
2, 2
2, 1
4, 1
4, 4
6, 4
6, 1
```

1810.000000 ms

4

```
Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm
Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI?
1. File
2. CLI
Silakan masukkan nomor pilihan Anda!
>> 2
Masukkan Anda adalah dari CLI
Masukkan banyaknya token unik yang ingin dijadikan matrix sequence
>> -
Banyaknya token unik harus lebih dari 0!
Masukkan banyaknya token unik yang ingin dijadikan matrix sequence
>> 6
Masukkan token unik yang ingin dijadikan matrix sequence
AA BB CC DD EE FFF
Panjang token harus 2! Silakan masukkan ulang token!
FF
Masukkan ukuran buffer yang diinginkan
>> 0
Ukuran buffer harus lebih dari 0!
Masukkan ukuran buffer yang diinginkan
>> 6
Masukkan ukuran matrix sequence yang diinginkan
>> 5 5
Masukkan banyaknya sequence yang diinginkan
>> 3
Masukkan ukuran maksimal sequence yang diinginkan
>> 3

Berikut merupakan matrix sequence random
DD EE BB BB
BB FF FF FF EE
DD BB CC BB FF
DD FF EE FF CC
EE FF AA EE EE

Berikut merupakan sequence unik dan reward random
Sequence 1: BB FF dengan reward 30 dan panjang 2
Sequence 2: FF dengan reward -50 dan panjang 1
Sequence 3: EE dengan reward -20 dan panjang 1

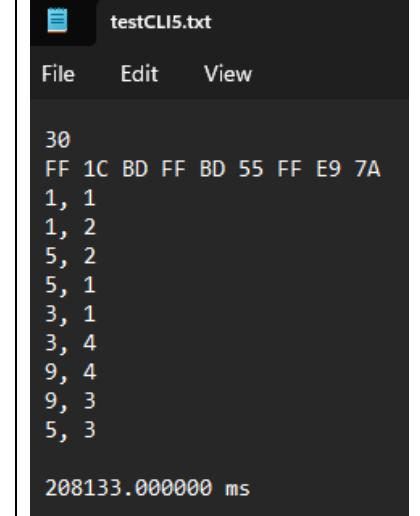
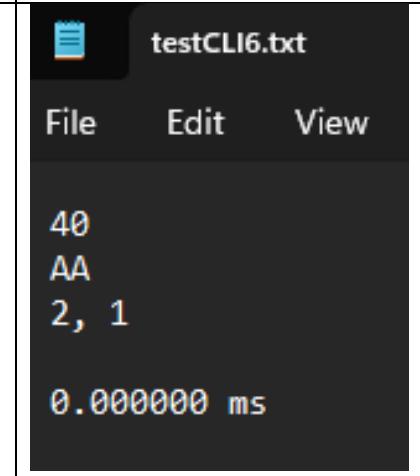
Tidak ada solusi!
1.000000 ms
Apakah Anda ingin menyimpan solusi ke dalam file .txt?
1. Ya
2. Tidak
Silakan masukkan nomor pilihan Anda!
>> 1
Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file)
>> testCLI4.txt
```

testCLI4.txt

File Edit View

Tidak ada solusi!

1.000000 ms

5	<pre>Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI? 1. File 2. CLI Silakan masukkan nomor pilihan Anda! >> 2 Masukan Anda adalah dari CLI Masukkan banyaknya token unik yang ingin dijadikan matrix sequence >> 6 Masukkan token unik yang ingin dijadikan matrix sequence 1C 55 7A BD E9 FF Masukkan ukuran buffer yang diinginkan >> 9 Masukkan ukuran matrix sequence yang diinginkan >> 10 10 Masukkan banyaknya sequence yang diinginkan >> 6 Masukkan ukuran maksimal sequence yang diinginkan >> 5 Berikut merupakan matrix sequence random FF FF BD 55 FF FF 7A 1C 7A 1C 1C 55 E9 BD BD 55 FF 7A FF BD 55 BD FF BD 7A 7A 7A BD E9 1C 55 FF 55 1C 7A BD BD FF 55 FF 1C 55 BD FF 55 1C 55 7A E9 E9 7A 55 FF 1C FF BD FF E9 E9 E9 55 7A E9 55 7A 7A BD BD E9 7A FF 7A 1C BD 1C 1C 7A BD 1C FF 55 FF BD FF 1C 1C BD 1C E9 7A 1C E9 BD 7A BD 7A 55 BD 1C Berikut merupakan sequence unik dan reward random Sequence 1: 55 BD E9 7A dengan reward 5 dan panjang 4 Sequence 2: BD 55 FF E9 7A dengan reward 45 dan panjang 4 Sequence 3: 55 FF 55 dengan reward -40 dan panjang 3 Sequence 4: BD FF BD 55 FF dengan reward 15 dan panjang 5 Sequence 5: E9 dengan reward -30 dan panjang 1 Sequence 6: 1C 7A 1C 55 dengan reward -35 dan panjang 4 30 FF 1C BD FF BD 55 FF E9 7A 1, 1 1, 2 5, 2 5, 1 3, 1 3, 4 9, 4 9, 3 5, 3 208133.000000 ms Apakah Anda ingin menyimpan solusi ke dalam file .txt? 1. Ya 2. Tidak Silakan masukkan nomor pilihan Anda! >> 1 Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file) >> testCLI5.txt </pre>	 <p>testCLI5.txt</p> <p>File Edit View</p> <p>30</p> <p>FF 1C BD FF BD 55 FF E9 7A</p> <p>1, 1 1, 2 5, 2 5, 1 3, 1 3, 4 9, 4 9, 3 5, 3</p> <p>208133.000000 ms</p>
6	<pre>Cyberpunk 2077 Breach Protocol Solver with Brute Force Algorithm Apakah Anda ingin memasukkan matrix dan sequence dari file atau dari CLI? 1. File 2. CLI Silakan masukkan nomor pilihan Anda! >> 2 Masukan Anda adalah dari CLI Masukkan banyaknya token unik yang ingin dijadikan matrix sequence >> 2 Masukkan token unik yang ingin dijadikan matrix sequence AA BB Masukkan ukuran buffer yang diinginkan >> 1 Masukkan ukuran matrix sequence yang diinginkan >> 2 2 Masukkan banyaknya sequence yang diinginkan >> 2 Masukkan ukuran maksimal sequence yang diinginkan >> 2 Berikut merupakan matrix sequence random BB AA AA AA Berikut merupakan sequence unik dan reward random Sequence 1: AA dengan reward 40 dan panjang 1 Sequence 2: BB dengan reward 30 dan panjang 1 #0 AA 2, 1 0.000000 ms Apakah Anda ingin menyimpan solusi ke dalam file .txt? 1. Ya 2. Tidak Silakan masukkan nomor pilihan Anda! >> 1 Masukkan nama file yang diinginkan (maksimal 200 karakter, jangan lupa tambahkan .txt pada akhir nama file) >> testCLI6.txt </pre>	 <p>testCLI6.txt</p> <p>File Edit View</p> <p>40</p> <p>AA</p> <p>2, 1</p> <p>0.000000 ms</p>

Referensi

Munir, R. (2024). *Algoritma Brute Force (Bagian 1)*. Diakses pada 12 Februari 2024, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf)

Link menuju *repository* Github:

https://github.com/nicholasrs05/Tucil1_13522144

Lampiran

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		✓