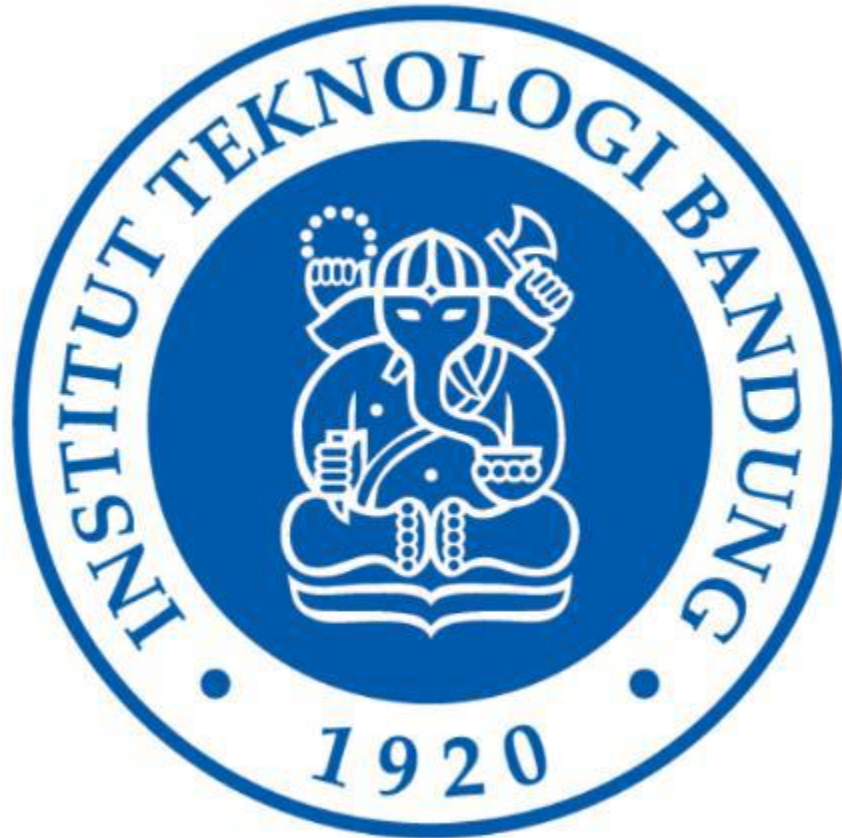


Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2023/2024

**Penyelesaian Cyberpunk 2077 Breach Protocol dengan
Algoritma Brute Force**



Eduardus Alvito Kristiadi 13522004

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

DAFTAR ISI

BAB I DESKRIPSI PERMASALAHAN	3
BAB II ALGORITMA BRUTE FORCE	4
BAB III SOURCE CODE	5
BAB IV UJI COBA PROGRAM	7
DAFTAR REFERENSI	8
LAMPIRAN	9
Tabel Checklist	9
Repository Github	9

BAB I

DESKRIPSI PERMASALAHAN

Cyberpunk 2077 adalah sebuah permainan video action RPG yang dikembangkan dan dipublikasikan oleh CD Projekt Red. Berlatar di sebuah dunia terbuka bernama Night City, pemain berperan sebagai tentara bayaran bernama V dalam perspektif kamera orang pertama. Permainan tersebut dirilis untuk Microsoft Windows, PlayStation 4, Stadia, dan Xbox One pada 10 Desember 2020, dan untuk PlayStation 5 dan Xbox Series X/S pada 2021.

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 horizontal, 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.
6. Sekuens memiliki panjang minimal berupa dua token.

BAB II

ALGORITMA BRUTE FORCE

Algoritma brute force adalah pendekatan yang lempang (*straightforward*) untuk memecahkan suatu persoalan. Biasanya algoritma brute force didasarkan pada: pernyataan pada persoalan problem statement dan definisi konsep yang dilibatkan. Algoritma brute force memecahkan persoalan dengan sangat sederhana, secara langsung, dan jelas. Pada persoalan Cyberpunk 2077 Breach Protocol ini, program akan mencari seluruh kemungkinan kombinasi sequence yang mungkin untuk mendapatkan poin tertinggi. Jika ditemukan adanya sequence yang sesuai, memenuhi buffer size dan memiliki poin tertinggi, maka itu adalah solusi. Jika tidak ada sequence yang sesuai entah dari kombinasi token, maupun dari buffer size tidak sesuai, maka dianggap tidak ada solusi.

BAB III

SOURCE CODE

```
with open('test\mboh.txt', 'r') as file:
    lines = [line.strip() for line in file.readlines()] # Strip whitespace

    # Extract and convert buffer size
    buffer_size = int(lines[0])

    # Extract and convert matrix dimensions
    matrix_width, matrix_height = map(int, lines[1].split())

    # Extract matrix
    matrix_start = 2
    matrix_end = matrix_start + matrix_height
    matrix = lines[matrix_start:matrix_end]

    # Extract and convert number of sequences
    number_of_sequences_start = matrix_end
    number_of_sequences = int(lines[number_of_sequences_start])

    # Extract sequences and rewards
    sequences = []
    current_line = number_of_sequences_start + 1
    for _ in range(number_of_sequences):
        sequence = lines[current_line]
        reward = int(lines[current_line + 1])
        sequences.append((sequence, reward))
        current_line += 2 # Move to the next sequence-reward pair

# Writing to a new file
with open('test\processed_mboh.txt', 'w') as file:
    file.write("Buffer Size: " + str(buffer_size) + '\nmatrix_height')
    file.write("Matrix Width and Height: " + str(matrix_width) + " " + str(matrix_height) +
'\nmatrix_height')
    file.write("Matrix:\nmatrix_height")
    for row in matrix:
        file.write(row + '\nmatrix_height')
    file.write("Number of Sequences: " + str(number_of_sequences) + '\nmatrix_height')
    file.write("Sequences and Rewards:\nmatrix_height")
    for seq, reward in sequences:
        file.write(seq + ' ' + str(reward) + '\nmatrix_height')
```

```

sec2 = [ 1, 2 ]

for i in range (0,4):
    for j in range (0,4):
        print(tes[i][j], end=" ")
    print()

print()

lst = []
sisi = 4
buffer_size = 4
x = 0
y = 0
peh = 0
tempY = []
Xsemu = 0
Ysemu = 0
Xtemp = 0
Ytemp = 0
trapY = "mbuh"
trapX = "mbuh"
indicator = 0
while (x < matrix_height):
    if (len(lst) == 0):
        lst.append((y,x,tes[y][x]))
    if (len(lst) > 0):
        if (len(lst) % 2 != 0 and len(lst) < buffer_size):
            # bawah
            if (trapY == "mbuh" or trapY == "bawah"):
                for vert in range (Ysemu+1,sisi,1):
                    lst.append((vert,Xsemu,tes[vert][Xsemu]))
                    Ysemu = vert
                    Xtemp = Xsemu
                    print(lst)
                    break
            # atas
            elif (trapY == "mbuh" or trapY == "atas"):
                for vert in range (Ysemu-1,-1,-1):
                    lst.append((vert,Xsemu,tes[vert][Xsemu]))
                    Ysemu = vert
                    Xtemp = Xsemu
                    print(lst)
                    break

        if (len(lst) % 2 == 0 and len(lst) < buffer_size):
            # kanan
            if (trapX == "mbuh" or trapX == "kanan"):
                for hor in range (Xsemu+1,sisi,1):
                    lst.append((Ysemu,hor,tes[Ysemu][hor]))
                    Xsemu = hor
                    Ytemp = Ysemu
                    print(lst)
                    break
            # kiri
            elif (trapX == "mbuh" or trapX == "kiri"):
                for hor in range (Xsemu-1,-1,-1):
                    lst.append((Ysemu,hor,tes[Ysemu][hor]))
                    Xsemu = hor

```

BAB IV

UJI COBA PROGRAM

```
test > mboh.txt
1 4
2 4 4
3 A B C 1
4 D E F 2
5 G H I 3
6 J K L 4
7 3
8 BD E9 1C
9 15
10 BD 7A BD
11 20
12 BD 1C BD 55
13 30
```

```
PS D:\OneDrive - Institut Teknologi Bandung\COLLEGE\SEMESTER 4\Strategi Algoritma\Tucil\Tucil1\Tucil1_13522004> cd /d "d:\OneDrive - Institut Teknologi Bandung\COLLEGE\SEMESTER 4\Strategi Algoritma\Tucil\Tucil1\Tucil1_13522004"; & "d:\EDUARD\Python3\python.exe" "c:\Users\Eduard\vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy_launcher" "59872" "--" "D:\OneDrive - Insti
tut Teknologi Bandung\COLLEGE\SEMESTER 4\Strategi Algoritma\Tucil\Tucil1\Tucil1_13522004\src\main.py"
A B C 1
D E F 2
G H I 3
J K L 4

[(0, 0, 'A'), (1, 0, 'D')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E'), (2, 1, 'H')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E'), (3, 1, 'K')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E'), (0, 1, 'B')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E'), (2, 1, 'H')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 1, 'E'), (3, 1, 'K')]
[(0, 0, 'A'), (1, 0, 'D')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 2, 'F'), (2, 2, 'I')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 2, 'F'), (3, 2, 'L')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 2, 'F'), (0, 2, 'C')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 2, 'F'), (2, 2, 'I')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 2, 'F'), (3, 2, 'L')]
[(0, 0, 'A'), (1, 0, 'D')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 3, '2'), (2, 3, '3')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 3, '2'), (3, 3, '4')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 3, '2'), (0, 3, '1')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 3, '2'), (2, 3, '3')]
[(0, 0, 'A'), (1, 0, 'D'), (1, 3, '2'), (3, 3, '4')]
[(0, 0, 'A'), (1, 0, 'D')]
PS D:\OneDrive - Institut Teknologi Bandung\COLLEGE\SEMESTER 4\Strategi Algoritma\Tucil\Tucil1\Tucil1_13522004> []
```

Program belum berhasil 100%, namun hanya bisa mencetak setiap kombinasi setiap buffer yang ada sesuai dengan buffer size.

DAFTAR REFERENSI

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2023-2024/stima23-24.htm>

LAMPIRAN

No.	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		✓

Repository Github

https://github.com/Edvardesu/Tucil1_13522004