

Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Penyelesaian Permainan Breach Protocol dengan Algoritma Brute Force



Disusun oleh:

Muhamad Rafli Rasyiidin 13522088

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

Daftar Isi

Daftar Isi	2
BAB 1	
Algoritma Brute Force	3
BAB 2	
Source Program	4
BAB 3	
Tangkapan Layar Hasil Pengujian	13
Link Repository	25
Lampiran	25

BAB 1

Algoritma Brute Force

Algoritma *Brute Force* merupakan pendekatan yang lempang (straightforward) untuk memecahkan persoalan. Algoritma yang digunakan dalam penyelesaian permainan Breach Protocol dengan pendekatan *brute force* adalah sebagai berikut:

1. Pilih salah satu token pada baris pertama
2. Pilih salah satu token yang memiliki posisi kolom yang sama dengan token yang telah dipilih sebelumnya. Pastikan posisi token yang dipilih saat ini belum pernah digunakan pada pemilihan token sebelum-sebelumnya
3. Kemudian, pilih lagi salah satu token yang memiliki posisi baris yang sama dengan token yang telah dipilih sebelumnya. Pastikan posisi token yang dipilih saat ini belum pernah digunakan pada pemilihan token sebelum-sebelumnya
4. Ulangi langkah kedua dan ketiga secara berurutan sehingga jumlah token yang dipilih sama dengan panjang maksimal buffer. Lakukan pengecekan sekuens setiap kali mengulangi langkah kedua dan ketiga
5. Jika panjang maksimal buffer telah terpenuhi, mundur 1 langkah (tandai banyak langkah setelah mundur sebagai *mark*) dan pilih token lain yang belum pernah dipilih sebelumnya (mengubah token pada langkah setelah *mark* dengan token lain sesuai dengan baris/kolom yang sama). Sesuaikan pemilihan token dengan pola vertikal-horizontal seperti pada langkah kedua dan ketiga
6. Ulangi langkah kelima hingga tidak ada lagi kemungkinan kombinasi yang tersisa
7. Setelah tidak ada lagi kemungkinan yang tersisa, mundur 1 langkah dari *mark* dan lakukan langkah kedua, ketiga, keempat, dan kelima. Lakukan hingga tidak ada kemungkinan kombinasi yang tersisa
8. Lakukan langkah ketujuh setiap kali tidak ada kemungkinan kombinasi yang tersisa hingga akhirnya *mark* berada pada baris pertama
9. Setelah *mark* berada pada baris pertama, pilih token lain yang berada pada baris pertama dan ulangi langkah-langkah yang sama seperti token yang dipilih pada baris pertama sebelumnya
10. Lakukan langkah kesembilan hingga seluruh token pada baris pertama telah dipilih dan tidak ada lagi kombinasi yang tersisa

BAB 2

Source Program

Program ini ditulis dalam bahasa Python. Ada 3 *library* yang digunakan pada program ini, yaitu *library time*, *random*, dan *os*. Berikut merupakan *source code* dari program ini:

```
import time
import random
import os

def solve(coordinat, length, vertical):
    if length < buffer_size:
        if vertical:
            for i in range(row):
                if (i+1, coordinat+1) not in temp_coordinat:
                    temp_sequence.append(m[i][coordinat])
                    temp_coordinat.append((i+1, coordinat+1))
                    compare_sequence(temp_sequence)
                    solve(i, length+1, False)
                    temp_sequence.pop()
                    temp_coordinat.pop()
        else:
            for i in range(col):
                if (coordinat+1, i+1) not in temp_coordinat:
                    temp_sequence.append(m[coordinat][i])
                    temp_coordinat.append((coordinat+1, i+1))
                    compare_sequence(temp_sequence)
                    solve(i, length+1, True)
                    temp_sequence.pop()
                    temp_coordinat.pop()
```

```

def compare_sequence(sequence):
    global count
    global max_point
    global sequence_solution
    global coordinat_solution
    if check_sequence(sequence):
        if point > max_point:
            max_point = point
            sequence_solution = []
            coordinat_solution = []
            for i in range (len(sequence)):
                sequence_solution.append(sequence[i])
                coordinat_solution.append(temp_coordinat[i])
        elif point == max_point:
            if len(sequence) < len(sequence_solution):
                max_point = point
                sequence_solution = []
                coordinat_solution = []
                for i in range (len(sequence)):
                    sequence_solution.append(sequence[i])
                    coordinat_solution.append(temp_coordinat[i])

    count += 1

```

```

def check_sequence(sequence):
    global point
    point = 0
    check2 = False
    for i in range(len(m_sequences)):
        check = False
        if len(sequence) >= len(m_sequences[i][0]):
            for j in range(len(sequence)):
                if sequence[j] == m_sequences[i][0][0]:
                    k = 0
                    check = True
                    while k < len(m_sequences[i][0]) and check:
                        if j+k < len(sequence):
                            if sequence[j+k] != m_sequences[i][0][k]:
                                check = False
                            else:
                                k += 1
                        else:
                            check = False
                    if check:
                        point += m_sequences[i][1]
                        check2 = True
                        break
    return check2

```



```

count = 0
max_point = 0
sequence_solution = []
coordinat_solution = []
programLoop = True
while programLoop:
    print()
    printBreachProtocol()
    print("Silakan pilih metode input:")
    print("1. File")
    print("2. CLI")
    print("3. Exit program")
    pick = input(">> ")
    print()

    if pick == "1":
        checkError = False
        checkReadedSequence = True
        nama_file = input(str("Masukkan nama file: "))
        print()
        path = '../test/' + nama_file
        while not os.path.exists(path):
            print("File tidak ditemukan, periksa kembali nama file!\n")
            nama_file = input(str("Masukkan nama file: "))
            print()
            path = '../test/' + nama_file
        fp = open(path, "r")
        start = time.time()
        try:
            buffer_size = int(fp.readline()) # read buffer size
        except ValueError:
            print("Error: Ukuran buffer tidak valid")
            checkError = True
        try:
            temp_line = fp.readline()
            col, row = int(temp_line.split()[0]), int(temp_line.split()[1])
            m = [[0 for i in range(col)] for j in range(row)]
        except ValueError:
            print("Error: Ukuran matriks tidak valid")
            checkError = True
        else:
            try:
                for i in range(col): # read Breach Protocol Matrix
                    temp_line = fp.readline()
                    col = 0

```



```

        for word in temp_line.split():
            if len(word) != 2:
                print("Error: panjang token pada matriks tidak valid")
                checkReadedSequence = False
            m[i][col] = word
            col += 1
    except:
        print("Error: Jumlah kolom tidak sesuai")
        checkError = True
    else:
        try:
            num_of_sequences = int(fp.readline().split()[0]) # read the sequence and the reward
        except ValueError:
            print("Error: jumlah sequence tidak valid")
            checkError = True
        else:
            m_sequences = []
            for i in range(num_of_sequences):
                temp_line = fp.readline()
                sequences = []
                for word in temp_line.split():
                    if len(word) != 2:
                        print("Error: panjang token pada sequence tidak valid")
                        checkReadedSequence = False
                    sequences.append(word)
                try:
                    m_sequences.append((sequences, int(fp.readline()))))
                except ValueError:
                    print("Error: nilai reward tidak valid")
                    checkError = True
            if not checkError and checkReadedSequence:
                pass
            else:
                print("Silakan periksa kembali file yang akan digunakan!")
                checkError = False
                checkReadedSequence = True
                fp.close()
                continue
# width = col
# height = row

temp_sequence = []
temp_coordinat = []

```

```

for i in range(col):
    temp_sequence.append(m[0][i])
    temp_coordinat.append((1, i+1))
    solve(i, 1, True)
    temp_sequence.pop()
    temp_coordinat.pop()

if max_point != 0:
    solution_output()
else:
    no_solution_output()

end = time.time()
td = (end-start)
print(f"\n{td*1000:.03f} ms\n")

print("Apakah Anda ingin menyimpan solusi? (y/n)")
status = False
while not status:
    choose = str(input(">> "))
    if choose == "y":
        save()
        status = True
    elif choose == "n":
        status = True
    else:
        print("Masukkan y atau n saja!")

fp.close()

elif pick == "2":
    checkReadedSequence = True
    try:
        number_of_tokens = int(input("Masukkan jumlah token: "))
        token_list = str(input("Masukkan token dalam 1 baris: ")).split()
        if len(token_list) != number_of_tokens:
            print()
            print("Error: banyak token tidak sesuai")
            print("Periksa kembali masukkan yang diberikan!")
            continue
        for i in range(number_of_tokens):
            if len(token_list[i]) != 2:
                print()
                print("Error: panjang token tidak valid")
                print("Periksa kembali masukkan yang diberikan!")

```

```

        checkReadedSequence = False
        break
    if not checkReadedSequence:
        continue
    buffer_size = int(input("Masukkan panjang buffer maksimal: "))
    matrix_size = str(input('Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): ')).split()
    number_of_sequence = int(input("Masukkan jumlah sequence: "))
    max_sequence_size = int(input("Masukkan panjang maksimal sequence: "))
except ValueError:
    print()
    print("Error: masukkan tidak valid")
    print("Periksa kembali masukkan yang diberikan!")
    continue
start = time.time()

row = int(matrix_size[0])
col = int(matrix_size[1])
m = [[random.choice(token_list) for i in range(col)] for j in range(row)]
print()
print("Matrix:")
for i in range(row):
    for j in range(col):
        print(m[i][j], end=" ")
    print()

m_sequences = []
print("\nSequence:")
for i in range(number_of_sequence):
    sequence_length = random.randint(2, max_sequence_size)
    sequence = []
    for j in range(sequence_length):
        sequence.append(random.choice(token_list))
    reward = random.randint(0, 100)
    m_sequences.append((sequence, reward))
    for element in m_sequences[i][0]:
        print(element, end=" ")
    print(f"({m_sequences[i][1]})")
print()

temp_sequence = []
temp_coordinat = []

for i in range(col):
    temp_sequence.append(m[0][i])
    temp_coordinat.append((1, i+1))

```

```

        solve(i, 1, True)
        temp_sequence.pop()
        temp_coordinat.pop()

    if max_point != 0:
        solution_output()
    else:
        no_solution_output()

end = time.time()
td = (end-start)
print(f"\n{td:.03f} ms\n")

print("Apakah Anda ingin menyimpan solusi? (y/n)")
status = False
while not status:
    choose = str(input(">> "))
    if choose == "y":
        save()
        status = True
    elif choose == "n":
        status = True
    else:
        print("Masukkan y atau n saja!")
elif pick == "3":
    programLoop = False
else:
    print("Masukkan tidak valid!")

```

BAB 3

Tangkapan Layar Hasil Pengujian

Metode input:

1. File
 - a. Input:

```
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
```

Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88oooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~b. 88~b. 88 8b 88~b. 88
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y8888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~b. 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P
```

Silakan pilih metode input:

1. File
 2. CLI
 3. Exit program
- >> 1

Masukkan nama file: tc1.txt



Result

Maximum reward: 50

Token combination: 7A BD 7A BD 1C BD 55

(1,1)

(1,4)

(3,4)

(3,5)

(6,5)

(6,3)

(1,3)

250.225 ms

b. Input:

```
test > ≡ tc3.txt
1 7
2 6 6
3 7Y FG HU AB 7Y AB
4 FG HU 8U FG HU HU
5 AB 8U 7Y FG AB FG
6 HU FG FG 7Y HU 8U
7 HU AB HU 7Y HU AB
8 8U HU HU 7Y 8U HU
9 5
10 AB IF 8U AB HU
11 94
12 IF 8U 7Y AB HU AB
13 57
14 HU 7Y AB 7Y
15 12
16 HU 8U IF AB
17 43
18 8U 7Y IF
19 30
```


Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~b. 88~b. 88~b. 88~b. 88~b.
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~b. .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~b. 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 1

Masukkan nama file: tc3.txt



Result
Maximum reward: 12
Token combination: HU 7Y AB 7Y
(3,1)
(3,3)
(1,3)
(1,1)

432.702 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): tc3res.txt
Solusi berhasil disimpan!
```

c. Input:

```
test > ≡ tc4.txt
1 7
2 6 6
3 8E 7U UH 6Y 7U 7U
4 8E UH UH 7U 7U 6Y
5 8E 6Y 6Y 6Y 8E 7U
6 UH 8E 7U UH AB AB
7 6Y 7U UH AB 6Y 8E
8 7U UH AB AB 8E AB
9 4
10 6Y 7U
11 51
12 UH 3C AB
13 7
14 3C AB
15 2
16 7U 7U 8E
17 94
```

Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~bbb. 88`8b 88~bbb 88~bbb 88 8b 88~bbb
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 `YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~88' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~bbb 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 `YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 1

Masukkan nama file: tc4.txt

F O U T I D

Result
Maximum reward: 145
Token combination: 6Y 7U 7U 8E
(4,1)
(4,2)
(5,2)
(5,3)

337.158 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): tc4res.txt

Solusi berhasil disimpan!
```

d. Input:

```
1 4
2 6 6
3 UF DO A9 B3 74 31
4 DD 7Y DB L4 QD 8A
5 BG 61 W7 QN TJ 91
6 C8 UO T0 PA UA D4
7 CQ 11 R2 DQ KN CF
8 98 IQ TL AX 9I 8E
9 4
10 AB UO AB UO AB
11 59
12 UF 7Y 9I 9I
13 6
14 BG UF
15 -50
16 UF AB
17 92
18 9I 9I 7Y BG
19 3
```

Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88oooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~ 88~ 88 8b 88~88
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y8888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. ~88~ .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~ 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P
```

```
Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 1
```

Masukkan nama file: tc2.txt



```
Result
Maximum reward: 0
No solution was found!
```

1.999 ms

```
Apakah Anda ingin menyimpan solusi? (y/n)
>> n
```


e. Input:

```
test > data7.txt
1 8
2 8 8
3 K9 D7 0F 7B 7B 15 7B J8
4 FF K9 0F FF 7B D7 7B K9
5 0F 7B 15 00 FF FF 15 15
6 0F FF J8 J8 D7 15 J8 K9
7 J8 0F K9 0F 15 7B D7 J8
8 00 K9 0F 00 0F FF 7B J8
9 J8 D7 K9 K9 00 00 15 FF
10 7B 00 7B 0F 7B 15 00 15
11 6
12 J8 FF 15 J8 K9
13 54
14 K9 K9 00 7B 7B
15 -30
16 15 7B J8 K9 7B 7B
17 -63
18 15 FF D7 D7
19 6
20 K9 15 FF J8 7B FF
21 -45
22 D7 FF K9
23 34
```

Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~~~~b. 88`8b 88~~~~ 88~~~~88 8b 88~~~~88
88 `8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~~~~ 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 1

Masukkan nama file: data7.txt

F O U N D

Result
Maximum reward: 88
Token combination: D7 FF K9 J8 FF 15 J8 K9
(2,1)
(2,4)
(8,4)
(8,6)
(6,6)
(6,1)
(8,1)
(8,2)

72109.897 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): data7res.tct
```

f. Input:

```
test > tc5.txt
1 7
2 8 8
3 88 03 03 G7 PO 88 PO GG
4 B3 03 03 C4 HA 03 03 88
5 GG B3 PO 03 C4 88 B3 03
6 B3 HA LO LO GG 88 88 LO
7 C4 03 HA HA JK G7 PO B3
8 GG 03 C4 B3 88 PO G7 JK
9 03 JK 03 JK JK LO 88 B3
10 B3 C4 LO G7 LO 88 GG C4
11 5
12 GG HA HA G7
13 49
14 03 GG
15 33
16 HA HA
17 46
18 HA B3
19 64
20 PO 88
21 33
```


Output:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88oooo 88ooo88 8P 88ooo88
88~~~~b. 88`8b 88~~~~ 88~~~~88 8b 88~~~~88
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~~~~ 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 1

Masukkan nama file: tc5.txt



Result
Maximum reward: 143
Token combination: 03 HA HA B3 PO 88
(3,1)
(3,5)
(4,5)
(4,6)
(6,6)
(6,1)

4697.662 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> n
```

2. CLI

a. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~b. 88~b. 88~b. 88~b. 88~b.
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~b. 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 5
Masukkan token dalam 1 baris: BD 1C 7A 55 E9
Masukkan panjang buffer maksimal: 7
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 6 6
Masukkan jumlah sequence: 3
Masukkan panjang maksimal sequence: 4
```

Output:

```
Matrix:
BD 55 7A E9 55 55
1C 55 E9 55 E9 E9
E9 E9 BD BD 55 BD
7A 55 55 55 BD 55
E9 7A 7A 55 E9 BD
BD 7A BD BD 1C 55

Sequence:
E9 E9 1C (48)
E9 BD E9 (75)
7A 1C 1C 1C (9)

      [E] [C] [U] [N] [D]

Result
Maximum reward: 123
Token combination: BD E9 BD E9 E9 1C
(1,1)
(1,3)
(3,3)
(3,2)
(5,2)
(5,6)

0.287 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): random1res.txt

Solusi berhasil disimpan!
```

b. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~b. 88~b. 88 8b 88~b.
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~b. 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 6
Masukkan token dalam 1 baris: IF 22 11 ST IM AA
Masukkan panjang buffer maksimal: 7
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 8 8
Masukkan jumlah sequence: 5
Masukkan panjang maksimal sequence: 4
```

Output:

```
Matrix:
IF 22 IF IF AA 22 AA ST
AA ST 11 AA AA AA 22 IM
11 AA AA IM IF IF ST AA
22 22 IM 22 IF AA ST IF
11 IM 11 AA ST ST 11 IM
IM IM 11 11 AA IF IM 22
11 IM 11 11 11 IF 11
IF IM IM IF 11 AA IM ST

Sequence:
11 IM (45)
AA ST ST 22 (31)
ST AA IF IF (84)
IF IF AA AA (47)
ST IF IF IM (100)

Result
Maximum reward: 145
Token combination: IF 11 IM ST IF IF IM
(1,1)
(1,5)
(8,5)
(8,1)
(4,1)
(4,8)
(2,8)

4.176 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): random2res.txt

Solusi berhasil disimpan!
```

c. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88oooo 88ooo88 8P 88ooo88
88~bbb. 88`8b 88~bbb 88~bbb 88 8b 88~bbb
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. ~~~88~~~ .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 88 8P 88 88 88
88~bbb 88`8b 88 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 8
Masukkan token dalam 1 baris: HE EH YA 13 03 PE JK 46
Masukkan panjang buffer maksimal: 7
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 10 10
Masukkan jumlah sequence: 5
Masukkan panjang maksimal sequence: 5
```

Output:

```
Matrix:
HE EH 46 03 JK EH HE 13 YA
13 EH HE HE 03 46 HE EH 46 HE
46 HE HE 03 13 JK EH PE 46 EH
HE EH HE PE JK 46 46 PE 03 YA
13 03 EH JK 46 PE HE YA JK YA
EH 03 EH PE HE HE 13 03 HE YA
EH JK EH JK HE HE 46 YA 13 46
03 JK 03 13 13 EH YA 13 EH
JK EH JK PE EH PE 46 PE 03 JK
03 13 13 03 46 HE 46 YA 13 EH

Sequence:
EH 46 HE (7)
03 JK HE HE HE (6)
YA 46 (15)
03 HE PE PE JK (93)
PE YA YA EH (63)

Result
Maximum reward: 108
Token combination: 03 HE PE PE JK YA 46
(5,1)
(5,6)
(4,6)
(4,9)
(10,9)
(10,1)
(3,1)

21.529 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): random3res.txt
Solusi berhasil disimpan!
```

d. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88'8b 88~ 88~88 8b 88~88
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y8888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~ 88'8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 5
Masukkan token dalam 1 baris: AM PM 12 TY UI
Masukkan panjang buffer maksimal: 7
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 7 7
Masukkan jumlah sequence: 4
Masukkan panjang maksimal sequence: 5
```

Output:

```
Matrix:
12 AM PM TY PM AM AM
AM PM AM TY 12 UI 12
PM 12 TY TY TY UI AM
PM AM 12 UI AM 12 12
AM UI TY TY TY AM PM
UI TY UI PM 12 12 PM
UI UI UI PM TY AM UI

Sequence:
12 PM UI AM (72)
12 AM (80)
12 TY (88)
PM TY 12 TY (76)

F O U N D

Result
Maximum reward: 244
Token combination: 12 AM PM TY 12 TY
(1,1)
(1,2)
(2,2)
(2,6)
(5,6)
(5,3)

1.215 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): result_random4.txt

Solusi berhasil disimpan!
```

e. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~b. 88~b. 88 8b 88~b.
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y88888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~b. 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y88888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 10
Masukkan token dalam 1 baris: AA A1 BB B2 CC C3 DD D4 EE E5
Masukkan panjang buffer maksimal: 7
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 3 3
Masukkan jumlah sequence: 2
Masukkan panjang maksimal sequence: 4
```

Output:

```
Matrix:
CC E5 C3
D4 DD A1
B2 AA A1

Sequence:
B2 E5 (93)
D4 AA (97)

NOT FOUND

Result
Maximum reward: 0
No solution was found!

0.011 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): result_random5.txt

Solusi berhasil disimpan!
```

f. Input:

```
d8888b. d8888b. d88888b .d8b. .o88b. db db
88 `8D 88 `8D 88' d8' `8b d8P Y8 88 88
88oooY' 88oobY' 88ooooo 88ooo88 8P 88ooo88
88~b. 88`8b 88~ 88~ 88~ 88 8b 88~ 88
88 8D 88 `88. 88. 88 88 Y8b d8 88 88
Y8888P' 88 YD Y8888P YP YP `Y88P' YP YP

d8888b. d8888b. .d88b. d888888b .d88b. .o88b. .d88b. db
88 `8D 88 `8D .8P Y8. `~88~' .8P Y8. d8P Y8 .8P Y8. 88
88oodD' 88oobY' 88 88 88 88 88 8P 88 88 88
88~ 88`8b 88 88 88 88 88 8b 88 88 88
88 88 `88. `8b d8' 88 `8b d8' Y8b d8 `8b d8' 88booo.
88 88 YD `Y88P' YP `Y88P' `Y88P' `Y88P' Y8888P

Silakan pilih metode input:
1. File
2. CLI
3. Exit program
>> 2

Masukkan jumlah token: 10
Masukkan token dalam 1 baris: AA A1 BB B2 CC C3 DD D4 EE E5
Masukkan panjang buffer maksimal: 6
Masukkan ukuran matriks dengan format "baris kolom" (contoh: 6 6): 10 11
Masukkan jumlah sequence: 5
Masukkan panjang maksimal sequence: 4
```

Output:

```
Matrix:
D4 EE C3 C3 DD C3 C3 DD CC EE C3
B2 D4 E5 BB D4 E5 C3 A1 A1 E5 D4
BB BB EE B2 CC EE D4 A1 D4 DD AA
B2 BB DD DD D4 E5 A1 AA BB BB EE
BB CC B2 EE A1 D4 E5 E5 CC CC EE
DD BB EE B2 BB C3 DD C3 D4 AA AA
EE D4 A1 CC E5 DD DD C3 C3 DD A1
B2 AA BB AA AA CC D4 D4 B2 EE D4
E5 B2 E5 D4 BB C3 AA AA AA CC D4
DD B2 E5 E5 EE A1 AA AA EE E5 EE

Sequence:
A1 DD BB (98)
D4 C3 D4 C3 (62)
BB C3 CC (64)
CC A1 (34)
EE EE (91)

FOUNID

Result
Maximum reward: 223
Token combination: EE EE CC A1 DD BB
(10,1)
(10,8)
(6,8)
(6,10)
(1,10)
(1,3)

2.816 ms

Apakah Anda ingin menyimpan solusi? (y/n)
>> y
Masukkan nama file (contoh: hasil.txt): result_random6.txt

Solusi berhasil disimpan!
```


Link Repository

https://github.com/MRafliRasyiidin/Tucil1_13522088

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		✓