LAPORAN TUGAS KECIL

IF2211 - STRATEGI ALGORTIMA

“Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Brute Force”

Dosen:

Ir. Rila Mandala, M.Eng., Ph.D.

Monterico Adrian, S.T., M.T.

Nama :

Axel Santadi Warih (13522155)

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

SEMESTER II TAHUN 2023/2024

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Abstraksi

Brute force adalah metode pemecahan masalah komputasi yang sederhana namun kuat yang mencoba semua kemungkinan solusi secara sistematis. Dalam konteks algoritma, pendekatan brute force mengimplikasikan pengujian setiap kemungkinan kombinasi solusi untuk mencapai hasil yang diinginkan. Teknik ini bergantung pada kekuatan komputasi yang memadai untuk mengevaluasi semua kemungkinan solusi dalam waktu yang wajar. Brute force sering digunakan dalam pengembangan algoritma pencarian, penguraian sandi, dan pemecahan masalah optimasi di mana jumlah solusi mungkin terbatas atau terukur secara masuk akal.

Meskipun efektif, pendekatan brute force dapat menjadi tidak efisien dalam kasus di mana ruang pencarian sangat besar, sehingga memerlukan sumber daya komputasi yang substansial. Terlebih lagi, kelemahan utama dari pendekatan ini adalah kecenderungan untuk menjadi tidak praktis atau tidak efisien dalam situasi di mana ruang pencarian sangat besar, sehingga membuatnya kurang efektif dalam beberapa konteks aplikasi. Namun, dalam beberapa kasus, khususnya di mana alternatif solusi yang lebih efisien tidak tersedia atau kompleksitas permasalahan tidak terlalu besar, brute force tetap menjadi pilihan yang layak untuk menyelesaikan masalah yang rumit.

1.2 Penerapan Brute Force

Penerapan brute force dalam konteks pemecahan masalah komputasi menawarkan pendekatan yang sederhana namun kuat untuk menyelesaikan berbagai jenis masalah. Teknik ini mengharuskan sistem untuk mencoba semua kemungkinan solusi secara sistematis hingga menemukan solusi yang diinginkan. Brute force sering digunakan dalam berbagai bidang, termasuk kriptografi, optimasi, dan pengembangan perangkat lunak.

Penerapan Brute Force dalam Kriptografi

Dalam kriptografi, brute force digunakan untuk menguraikan sandi dengan mencoba semua kombinasi kunci yang mungkin hingga menemukan kunci yang benar. Misalnya, dalam enkripsi kunci simetris, brute force dapat diterapkan dengan mencoba setiap kombinasi kunci hingga pesan terdekripsi dengan benar.

Penerapan Brute Force dalam Algoritma Pencarian

Dalam algoritma pencarian, brute force dapat digunakan untuk mencari solusi optimal dengan mengevaluasi setiap kemungkinan solusi secara berurutan. Misalnya, dalam algoritma pencarian dalam graf, brute force mencoba semua jalur yang mungkin dari simpul awal ke simpul tujuan untuk menemukan jalur terpendek.

Penerapan Brute Force dalam Cyberpunk 2077 Breach Protocol

Dalam game Cyberpunk 2077, terdapat mekanisme yang disebut Breach Protocol di mana pemain harus memecahkan serangkaian kode untuk mengakses sistem keamanan. Brute force dapat diterapkan dalam Breach Protocol dengan mencoba semua kombinasi kode yang mungkin secara berurutan hingga menemukan kombinasi yang benar untuk mengakses sistem tersebut. Kode dapat diawali dengan mencari semua sekuens yang mungkin dari matrix tersebut, lalu program dapat mensortir sekuens - sekuens tersebut yang menganut semua aturan permainan, setelah itu kode dapat menghitung sekuens hasil yang menghasilkan poin paling banyak.

BAB II
SOURCE PROGRAM

2.1 readFile.py

```python
class Tucil:
    def __init__(self, buffer, row, col, jumlah_sequence, sequence_length, matriks, sequence, reward_sequence):
        self.buffer = buffer
        self.row = row
        self.col = col
        self.jumlah_sequence = jumlah_sequence
        self.sequence_length = sequence_length
        self.matriks = matriks
        self.sequence = sequence
        self.reward_sequence = reward_sequence

def baca_file(nama_file):
    with open(nama_file, 'r') as file:
        data = file.readlines()
    return data

def proses_data(data):
    buffer_size = int(data[0].strip())

    matrix_info = list(map(int, data[1].strip().split()))
    matrix_width = matrix_info[0]
    matrix_height = matrix_info[1]

    matrix = [list(map(str, line.strip().split())) for line in data[2:2+matrix_height]]

    num_sequences = int(data[2+matrix_height].strip())
    sequences = []
    rewards = []

    for i in range(num_sequences):
        sequences = [list(map(str, line.strip().split())) for line in data[3+matrix_height:3+matrix_height+(2*num_sequences):2]]
        rewards.append(int(data[2+matrix_height+i*2+2].strip()))
    sequence_length = max(len(seq) for seq in sequences)

    tucil = Tucil(buffer_size, matrix_width, matrix_height, num_sequences, sequence_length, matrix, sequences, rewards)

    return tucil
```

2.2 cli.py

```python
import random
import time

def countWords(str):
    return len(str.split())

def generateMatriks(tucil, token, jumlah_token_unik):
    random.seed(time.time())
    matriks = [[token[random.randint(0, jumlah_token_unik-1)] for _ in range(tucil.col)] for _ in range(tucil.row)]
    return matriks

def generateSequence(tucil, token, jumlah_token_unik):
    random.seed(time.time())
    sequence = [[token[random.randint(0, jumlah_token_unik-1)] for _ in range(random.randint(2, tucil.sequence_length))] for _ in range(tucil.jumlah_sequence)]
    return sequence

def generateRewardSequence(tucil):
    random.seed(time.time())
    reward_sequence = [random.randint(0, 99) for _ in range(tucil.jumlah_sequence)]
    return reward_sequence

def checkData(tucil):
    print("\nBerikut adalah data yang dihasilkan: \n\n")
    print("Buffer size: ", tucil.buffer)
    print("Row matriks: ", tucil.row)
    print("Col matriks: ", tucil.col)
    print("\nMatriks: ")
    for i in range(tucil.row):
        for j in range(tucil.col):
            print(tucil.matriks[i][j], end=" ")
        print()
    print("\nJumlah sequence: ", tucil.jumlah_sequence)
    print("Sequence length: ", tucil.sequence_length)
    print("\nSequences: ")
    for i in range(tucil.jumlah_sequence):
        print("Sequence ", i+1, ": ", end="")
        for j in range(len(tucil.sequence[i])):
            print(tucil.sequence[i][j], end=" ")
        print()
    print("\nReward sequences: ")
    for i in range(tucil.jumlah_sequence):
        print("Sequence ", i+1, ": ", tucil.reward_sequence[i])
    print()
```

2.3 brute.py

```python
class Point:
    def __init__(self, row, col):
        self.row = row
        self.col = col

def count_seq_length(tucil, idx):
    count = 0
    if idx < len(tucil.sequence):
        for i in range(min(tucil.sequence_length, len(tucil.sequence[idx]))):
            if tucil.sequence[idx][i] != "":
                count += 1
    return count

def count_score(tucil, temp):
    score = 0
    size = len(temp)
    for i in range(size):
        for j in range(tucil.jumlah_sequence):
            length = count_seq_length(tucil, j)
            if size - i >= length:
                count = 0
                for k in range(length):
                    if temp[i + k] == tucil.sequence[j][k]:
                        count += 1
                if count == length:
                    score += tucil.reward_sequence[j]
    if score == 0:
        score = -1
    return score

def same_sequence(route, betul):
    if len(route) != len(betul):
        return False
    else:
        for i in range(len(betul)):
            if route[i] != betul[i]:
                return False
        return True
```

```python
global max_route_coordinates

def search_route(tucil, matrix, row, col, route, route_points, visited, buffer, is_vertical):
    global max_reward, max_route, max_route_coordinates
    route.append(matrix[row][col])
    route_points.append(Point(row, col))
    visited[row][col] = True
    temp_reward = count_score(tucil, route)
    if temp_reward > max_reward:
        max_reward = temp_reward
        max_route = list(route)
        max_route_coordinates = list(route_points)  # update max_route_coordinates here
    if len(route) == buffer:
        route.pop()
        route_points.pop()
        visited[row][col] = False
        return
    if is_vertical:
        for i in range(tucil.row):
            if not visited[i][col]:
                search_route(tucil, matrix, i, col, route, route_points, visited, buffer, not is_vertical)
    else:
        for j in range(tucil.col):
            if not visited[row][j]:
                search_route(tucil, matrix, row, j, route, route_points, visited, buffer, not is_vertical)
    route.pop()
    route_points.pop()
    visited[row][col] = False

def write_sequence_to_file(filename, max_route, max_reward, max_route_coordinates, execution_time):
    with open(filename, 'w') as f:
        # Write max_reward
        f.write(str(max_reward) + '\n')

        # Write max_route
        f.write(' '.join(max_route) + '\n')

        # Write max_route_coordinates
        for point in max_route_coordinates:
            f.write(f'{point.row + 1}, {point.col + 1}\n')

        f.write('\n')

        # Write execution time
        f.write(str(execution_time) + ' ms\n')
```

BAB III
TESTCASE

3.1 Testcase 1

```
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155> cd bin
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin> python main.py
Apakah anda ingin memasukkan Input atau membaca .txt? (Input / Text / Gajadi): Text
Masukkan nama file yang ingin anda baca (akhiri dengan .txt): test.txt

Berikut adalah data yang dihasilkan:


Buffer size:  7
Row matriks:  6
Col matriks:  6

Matriks:
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

Jumlah sequence:  3
Sequence length:  4

Sequences:
Sequence  1 : BD E9 1C
Sequence  2 : BD 7A BD
Sequence  3 : BD 1C BD 55

Reward sequences:
Sequence  1 :  15
Sequence  2 :  20
Sequence  3 :  30

Max Reward:  50
Max Route: 7A BD 7A BD 1C BD 55
Max Route Coordinates: (1, 1) (4, 1) (4, 3) (5, 3) (5, 6) (3, 6) (3, 1)
Execution time:  1.2587299346923828  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
Masukkan nama file yang ingin anda simpan (akhiri dengan .txt): hasil1.txt
okay, file sudah tersimpan di folder test dengan nama hasil1.txt. Terima kasih sudah menggunakan program ini, sampai jumpa lagi :D .
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin>
```

3.2 Testcase 2

```
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin> python main.py
Apakah anda ingin memasukkan Input atau membaca .txt? (Input / Text / Gajadi): Input
Masukkan jumlah token unik: 5
Masukkan token apa saja yang akan dipakai: BD 1C 7A 55 E9
Masukkan jumlah buffer maksimal: 7
Masukkan ukuran matrix: 6 6
Masukkan jumlah sekuens: 3
Masukkan ukuran maksimal sekuens: 4

Berikut adalah data yang dihasilkan:


Buffer size:  7
Row matriks:  6
Col matriks:  6

Matriks:
BD 1C 1C BD E9 E9
E9 E9 7A BD 1C 7A
55 BD E9 1C 7A E9
7A E9 7A 7A BD 1C
BD 1C E9 7A 1C 7A
BD BD 55 1C 55 BD

Jumlah sequence:  3
Sequence length:  4

Sequences:
Sequence  1 : BD 1C 1C BD
Sequence  2 : E9 E9 E9 7A
Sequence  3 : BD 1C 7A 55

Reward sequences:
Sequence  1 :  90
Sequence  2 :  5
Sequence  3 :  96

Max Reward:  96
Max Route: BD E9 BD 1C 7A 55
Max Route Coordinates: (1, 1) (2, 1) (2, 4) (3, 4) (3, 5) (6, 5)
Execution time:  1.3028900623321533  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
Masukkan nama file yang ingin anda simpan (akhiri dengan .txt): hasil2.txt
okay, file sudah tersimpan di folder test dengan nama hasil2.txt. Terima kasih sudah menggunakan program ini, sampai jumpa lagi :D .
```

3.3 Testcase 3

```
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin> python main.py
Apakah anda ingin memasukkan Input atau membaca .txt? (Input / Text / Gajadi): Input
Masukkan jumlah token unik: 3
Masukkan token apa saja yang akan dipakai: A1 IP OD
Masukkan jumlah buffer maksimal: 5
Masukkan ukuran matrix: 5 5
Masukkan jumlah sekuens: 3
Masukkan ukuran maksimal sekuens: 3

Berikut adalah data yang dihasilkan:


Buffer size:  5
Row matriks:  5
Col matriks:  5

Matriks:
OD IP IP OD IP
IP OD IP OD IP
A1 OD OD A1 OD
OD OD OD OD IP
A1 A1 OD IP A1

Jumlah sequence:  3
Sequence length:  3

Sequences:
Sequence  1 : IP OD IP
Sequence  2 : OD IP OD
Sequence  3 : A1 OD OD

Reward sequences:
Sequence  1 :  83
Sequence  2 :  47
Sequence  3 :  58

Max Reward:  213
Max Route: IP OD IP OD IP
Max Route Coordinates: (1, 2) (2, 2) (2, 1) (1, 1) (1, 3)
Execution time:  0.01501321792602539  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
Masukkan nama file yang ingin anda simpan (akhiri dengan .txt): hasil3.txt
okay, file sudah tersimpan di folder test dengan nama hasil3.txt. Terima kasih sudah menggunakan program ini, sampai jumpa lagi :D .
```

3.4 Testcase 4

```
Masukkan token apa saja yang akan dipakai: AB CD EF GH IJ KL MN
Masukkan jumlah buffer maksimal: 10
Masukkan ukuran matrix: 6 6
Masukkan jumlah sekuens: 4
Masukkan ukuran maksimal sekuens: 5

Berikut adalah data yang dihasilkan:


Buffer size:  10
Row matriks:  6
Col matriks:  6

Matriks:
EF MN AB MN KL MN
GH MN IJ KL EF EF
EF MN AB AB EF MN
EF EF CD CD GH CD
AB EF MN GH EF CD
KL GH GH AB GH EF

Jumlah sequence:  4
Sequence length:  5

Sequences:
Sequence  1 : MN AB MN KL
Sequence  2 : MN IJ KL EF EF
Sequence  3 : MN AB AB EF
Sequence  4 : EF CD CD GH

Reward sequences:
Sequence  1 :  41
Sequence  2 :  1
Sequence  3 :  89
Sequence  4 :  60

Max Reward:  178
Max Route: MN MN AB AB EF MN AB AB EF
Max Route Coordinates: (1, 2) (3, 2) (3, 4) (6, 4) (6, 6) (1, 6) (1, 3) (3, 3) (3, 1)
Execution time:  199.08448767662048  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
```

3.5 Testcase 5

```
Ngomong opo toh mazzeh? Diketik lagi coba (Input / Text / Gajadi): 4
Ngomong opo toh mazzeh? Diketik lagi coba (Input / Text / Gajadi): 5
Ngomong opo toh mazzeh? Diketik lagi coba (Input / Text / Gajadi): Input
Masukkan jumlah token unik: 3
Masukkan token apa saja yang akan dipakai: AB CD EF
Masukkan jumlah buffer maksimal: 4
Masukkan ukuran matrix: 5 5
Masukkan jumlah sekuens: 3
Masukkan ukuran maksimal sekuens: 4

Berikut adalah data yang dihasilkan:


Buffer size:  4
Row matriks:  5
Col matriks:  5

Matriks:
CD CD CD AB CD
CD AB EF AB EF
AB EF CD CD CD
AB AB EF EF EF
CD AB CD EF CD

Jumlah sequence:  3
Sequence length:  4

Sequences:
Sequence  1 : CD CD AB
Sequence  2 : CD AB EF
Sequence  3 : EF AB

Reward sequences:
Sequence  1 :  32
Sequence  2 :  2
Sequence  3 :  56

Max Reward:  58
Max Route: CD AB EF AB
Max Route Coordinates: (1, 1) (3, 1) (3, 2) (2, 2)
Execution time:  0.003000020980834961  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
Masukkan nama file yang ingin anda simpan (akhiri dengan .txt): hasil5.txt
okay, file sudah tersimpan di folder test dengan nama hasil5.txt. Terima kasih sudah menggunakan program ini, sampai jumpa lagi :D .
```

3.6 Testcase 6

```
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin> python3 main.py
Apakah anda ingin memasukkan Input atau membaca .txt? (Input / Text / Gajadi): Input
Masukkan jumlah token unik: 6
Masukkan token apa saja yang akan dipakai: AB CD EF GH IJ KL
Masukkan jumlah buffer maksimal: 5
Masukkan ukuran matrix: 4 4
Masukkan jumlah sekuens: 3
Masukkan ukuran maksimal sekuens: 3

Berikut adalah data yang dihasilkan:


Buffer size:  5
Row matriks:  4
Col matriks:  4

Matriks:
EF KL GH GH
GH GH AB CD
KL EF CD KL
AB GH KL EF

Jumlah sequence:  3
Sequence length:  3

Sequences:
Sequence  1 : KL GH GH
Sequence  2 : GH AB CD
Sequence  3 : CD KL AB

Reward sequences:
Sequence  1 :  45
Sequence  2 :  86
Sequence  3 :  54

Max Reward:  86
Max Route: EF GH AB CD
Max Route Coordinates: (1, 1) (2, 1) (2, 3) (3, 3)
Execution time:  0.0039980411529541016  seconds
apakah hasil ini mau anda simpan ke dalam file? (Y/N): Y
Masukkan nama file yang ingin anda simpan (akhiri dengan .txt): hasil6.txt
okay, file sudah tersimpan di folder test dengan nama hasil6.txt. Terima kasih sudah menggunakan program ini, sampai jumpa lagi :D .
```

3.7 Testcase 7

```
PS C:\Users\Axel Santadi\Documents\Cool_Yeah\Tingkat_2\Semester_4\StiMa\Tucil\01\Tucil1_13522155\bin> python3 main.py
Apakah anda ingin memasukkan Input atau membaca .txt? (Input / Text / Gajadi): Gajadi
Oke deh masbro, see you next time :D .
```

BAB IV
LAMPIRAN

4.1 Link Repository Github

https://github.com/AxelSantadi/Tucil1_13522155.git

4.2 Tabel Laporan

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		✓