Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2023/2024

Penyelesaian Cyberpunk 2077 Breach Protocol dengan Algoritma Brute Force

Algoritma Brute Force

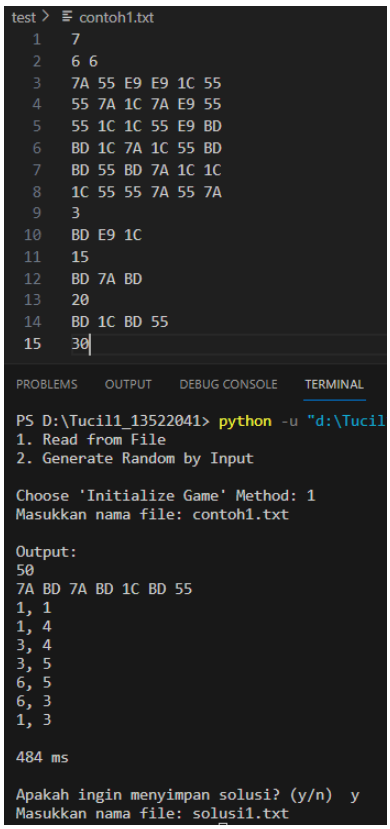
Program akan mengecek setiap jalur pada matriks dengan menggunakan metode DFS, yaitu melalui kolom atau baris terkecil terlebih dahulu dengan tetap mengikuti aturan pola horizontal dan vertikal bergantian. Pada setiap pengecekan jalur, program akan langsung mengecek nilai poin yang didapatkan pola tersebut dan membandingkannya dengan nilai poin terbesar sebelumnya dan sekaligus membandingkan panjang pola jika memiliki poin yang sama.

Source Code

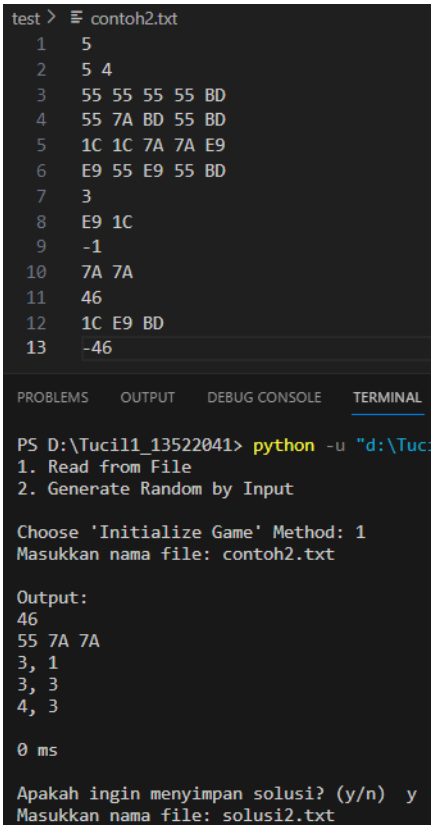
```python
import random
import time

def countPoint(arrPos):
    counter = 0
    arraySeqBaru = [x[0] for x in arrSeq]
    currSeq = [0 for x in arraySeqBaru]

    for i in range(len(arrPos)):
        for j in range(len(currSeq)):
            if currSeq[j] == "Done":
                continue
            if (matriks[arrPos[i][0]][arrPos[i][1]] == arraySeqBaru[j][currSeq[j]]):
                currSeq[j] += 1
                if currSeq[j] == len(arraySeqBaru[j]):
                    currSeq[j] = "Done"
                    counter += arrSeq[j][1]
            else:
                currSeq[j] = 0
    return counter

print("1. Read from File \n2. Generate Random by Input \n")
pilihan = input("Choose 'Initialize Game' Method: ")

if (pilihan == '1'):
    namaFile = input("Masukkan nama file: ")
    file = open("test/" + namaFile, "r")

    bufSize = int(file.readline())
    matSize = file.readline().split()

    matriks = []
    for i in range(int(matSize[1])):
        row = file.readline().split()
        matriks.append(row)

    nSeq = int(file.readline())
    arrSeq = []
    for i in range(nSeq):
        seq = file.readline().split()
        rew = int(file.readline())
        arrSeq.append([seq, rew])

    file.close()

elif (pilihan == '2'):
    nToken = input("Jumlah Token Unik: ")
    listToken = input("Token Unik: ").split()

    bufSize = int(input("Ukuran Buffer: "))
    matSize = input("Ukuran Matriks: ").split()

    matriks = []
    for i in range(int(matSize[1])):
        row = []
        for j in range(int(matSize[0])):
            elmt = random.choice(listToken)
            row.append(elmt)
        matriks.append(row)

    nSeq = int(input("Jumlah Sekuens: "))
    lenMaxSeq = int(input("Panjang Maksimal Sekuens: "))

    arrSeq = []
    for i in range(nSeq):
        lenSeq = random.randint(2, lenMaxSeq)
        seq = []
        while (seq == [] or [seq] in arrSeq): # Prevent same sequence to appear
            for j in range(lenSeq):
                elmt = random.choice(listToken)
                seq.append(elmt)
        arrSeq.append([seq])

    for i in range(nSeq):
        rew = random.randint(-50, 50)
        arrSeq[i].append(rew)

    print("\nSekuens: ")
    for i in arrSeq:
        print(i)
    print("\nMatriks: ")
    for i in matriks:
        print(i)
else:
    print("Invalid Input")
    exit()

arrSeq.sort(key=lambda x: x[1], reverse=True)

awal = time.time()
```

```python

maxPoint = 0
maxSeq = []

def bruteforce(arrP):
    global maxPoint
    global maxSeq
    arrPos = [a for a in arrP]
    for i in range(int(matSize[1])):
        currCol = arrPos[-1][1]
        if (len(arrPos) == bufSize):
            continue
        elif ([i, currCol] in arrPos):
            continue
        else:
            arrPos.append([i, currCol])
            poin = countPoint(arrPos)

            if (poin == maxPoint and len(maxSeq) > len(arrPos)):
                maxSeq = [a for a in arrPos]
            elif poin>maxPoint:
                maxPoint = poin
                maxSeq = [a for a in arrPos]

        for j in range(int(matSize[0])):
            if (len(arrPos) == bufSize):
                continue
            elif ([i, j] in arrPos):
                continue
            else:
                arrPos.append([i, j])
                poin = countPoint(arrPos)

                if (poin == maxPoint and len(maxSeq) > len(arrPos)):
                    maxSeq = [a for a in arrPos]
                elif poin>maxPoint:
                    maxPoint = poin
                    maxSeq = [a for a in arrPos]

                bruteforce(arrPos)

                arrPos.pop()

        arrPos.pop()

for i in range(int(matSize[0])):
    bruteforce([[0, i]])

akhir = time.time()

print("\nOutput: ")
strMaxSeq = ""
for [x,y] in maxSeq:
    strMaxSeq += (matriks[x][y] + " ")
strLocSeq = ""
for [x,y] in maxSeq:
    strLocSeq += (str(y+1)+", "+str(x+1)+"\n")

print(maxPoint)
print(strMaxSeq)
print(strLocSeq)

print(str(int((akhir - awal)*1000)) + " ms")

saveState = input("\nApakah ingin menyimpan solusi? (y/n)  ")
if (saveState == 'y'):
    filename = input("Masukkan nama file: ")
    file = open("test/"+filename, "w")
    file.write(str(maxPoint))
    file.write('\n')
    file.write(strMaxSeq)
    file.write('\n')
    file.write(strLocSeq)
    file.close()
```

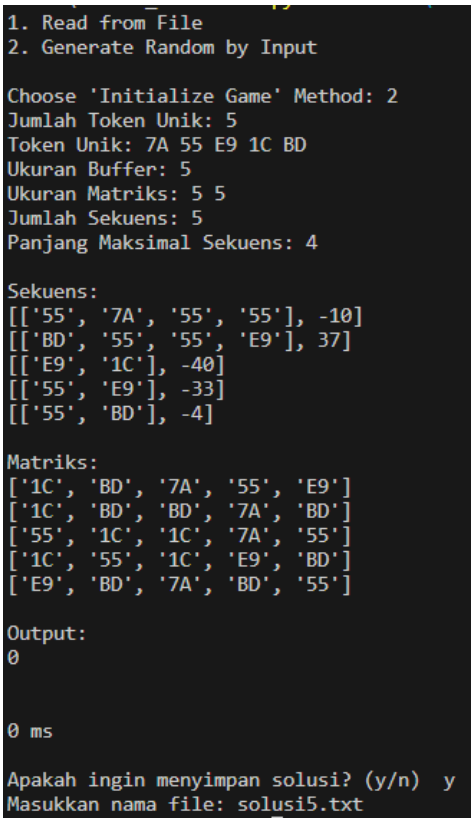
Contoh Hasil Eksekusi



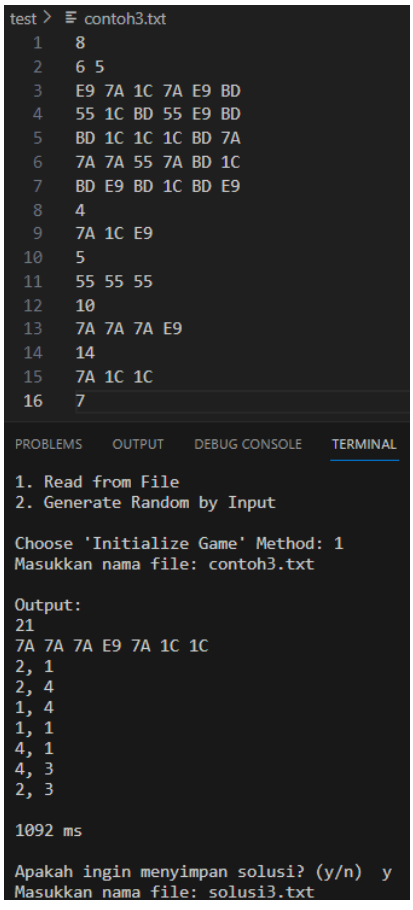
```
test > contoh1.txt
1  7
2  6 6
3  7A 55 E9 E9 1C 55
4  55 7A 1C 7A E9 55
5  55 1C 1C 55 E9 BD
6  BD 1C 7A 1C 55 BD
7  BD 55 BD 7A 1C 1C
8  1C 55 55 7A 55 7A
9  3
10 BD E9 1C
11 15
12 BD 7A BD
13 20
14 BD 1C BD 55
15 30

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Tucil1_13522041> python -u "d:\Tucil
1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 1
Masukkan nama file: contoh1.txt

Output:
50
7A BD 7A BD 1C BD 55
1, 1
1, 4
3, 4
3, 5
6, 5
6, 3
1, 3

484 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi1.txt
```



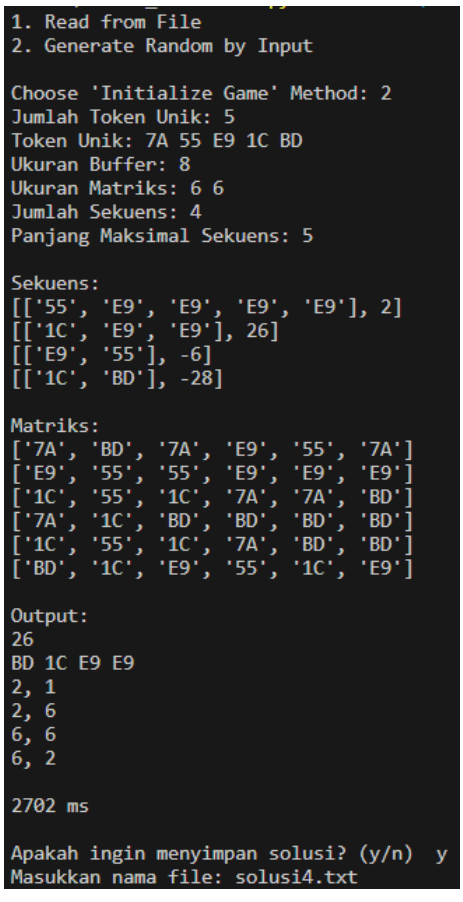
```
test > contoh2.txt
1  5
2  5 4
3  55 55 55 55 BD
4  55 7A BD 55 BD
5  1C 1C 7A 7A E9
6  E9 55 E9 55 BD
7  3
8  E9 1C
9  -1
10 7A 7A
11 46
12 1C E9 BD
13 -46

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\Tucil1_13522041> python -u "d:\Tuc
1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 1
Masukkan nama file: contoh2.txt

Output:
46
55 7A 7A
3, 1
3, 3
4, 3

0 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi2.txt
```



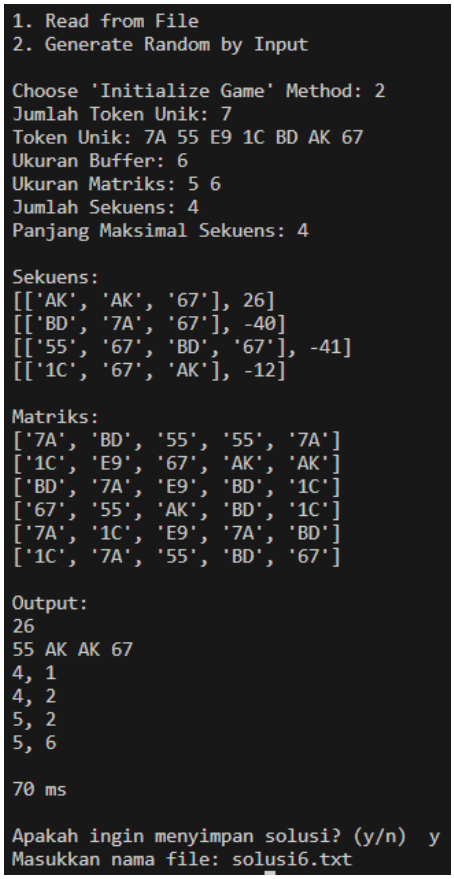
```
1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 2
Jumlah Token Unik: 5
Token Unik: 7A 55 E9 1C BD
Ukuran Buffer: 5
Ukuran Matriks: 5 5
Jumlah Sekuens: 5
Panjang Maksimal Sekuens: 4

Sekuens:
[['55', '7A', '55', '55'], -10]
[['BD', '55', '55', 'E9'], 37]
[['E9', '1C'], -40]
[['55', 'E9'], -33]
[['55', 'BD'], -4]

Matriks:
['1C', 'BD', '7A', '55', 'E9']
['1C', 'BD', 'BD', '7A', 'BD']
['55', '1C', '1C', '7A', '55']
['1C', '55', '1C', 'E9', 'BD']
['E9', 'BD', '7A', 'BD', '55']

Output:
0


0 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi5.txt
```



```
test > contoh3.txt
1  8
2  6 5
3  E9 7A 1C 7A E9 BD
4  55 1C BD 55 E9 BD
5  BD 1C 1C 1C BD 7A
6  7A 7A 55 7A BD 1C
7  BD E9 BD 1C BD E9
8  4
9  7A 1C E9
10 5
11 55 55 55
12 10
13 7A 7A 7A E9
14 14
15 7A 1C 1C
16 7

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 1
Masukkan nama file: contoh3.txt

Output:
21
7A 7A 7A E9 7A 1C 1C
2, 1
2, 4
1, 4
1, 1
4, 1
4, 3
2, 3

1092 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi3.txt
```



```
1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 2
Jumlah Token Unik: 5
Token Unik: 7A 55 E9 1C BD
Ukuran Buffer: 8
Ukuran Matriks: 6 6
Jumlah Sekuens: 4
Panjang Maksimal Sekuens: 5

Sekuens:
[['55', 'E9', 'E9', 'E9', 'E9'], 2]
[['1C', 'E9', 'E9'], 26]
[['E9', '55'], -6]
[['1C', 'BD'], -28]

Matriks:
['7A', 'BD', '7A', 'E9', '55', '7A']
['E9', '55', '55', 'E9', 'E9', 'E9']
['1C', '55', '1C', '7A', '7A', 'BD']
['7A', '1C', 'BD', 'BD', 'BD', 'BD']
['1C', '55', '1C', '7A', 'BD', 'BD']
['BD', '1C', 'E9', '55', '1C', 'E9']

Output:
26
BD 1C E9 E9
2, 1
2, 6
6, 6
6, 2

2702 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi4.txt
```



```
1. Read from File
2. Generate Random by Input

Choose 'Initialize Game' Method: 2
Jumlah Token Unik: 7
Token Unik: 7A 55 E9 1C BD AK 67
Ukuran Buffer: 6
Ukuran Matriks: 5 6
Jumlah Sekuens: 4
Panjang Maksimal Sekuens: 4

Sekuens:
[['AK', 'AK', '67'], 26]
[['BD', '7A', '67'], -40]
[['55', '67', 'BD', '67'], -41]
[['1C', '67', 'AK'], -12]

Matriks:
['7A', 'BD', '55', '55', '7A']
['1C', 'E9', '67', 'AK', 'AK']
['BD', '7A', 'E9', 'BD', '1C']
['67', '55', 'AK', 'BD', '1C']
['7A', '1C', 'E9', '7A', 'BD']
['1C', '7A', '55', 'BD', '67']

Output:
26
55 AK AK 67
4, 1
4, 2
5, 2
5, 6

70 ms

Apakah ingin menyimpan solusi? (y/n)  y
Masukkan nama file: solusi6.txt
```

Repository

https://github.com/Bana-man/Tucil1_13522041

AHMAD HASAN ALBANA (13522041)