

Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2020/2021

Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

Nama : Ahmad Saladin

NIM : 13519187

Kelas : K-04

Deskripsi Algoritma

1. Baca persoalan *cryptarithmic* dari file kemudian setiap operand dan hasil jawaban disimpan dalam bentuk list (rata kanan) yang panjangnya 10 karakter yang kemudian digabungkan menjadi matriks (matriks M). jika operand atau jawaban panjangnya kurang dari 10, indeks yang tidak terisi diisi dengan “” atau “”. Saat membaca file setiap kali huruf yang unik muncul akan disimpan dalam sebuah list (list A) yang panjangnya maksimal 10. Setiap huruf pertama dari tiap operan juga disimpan dalam list lain.
2. Lakukan enumerasi permutasi n buah angka dimana n adalah panjang list A yang akan ditempatkan dalam sebuah list (List B). List ini akan bersesuaian dengan list A, sehingga karakter pada indeks 0 pada list A akan diberi nilai yang berada pada indeks 0 list B dan begitu seterusnya. Setiap enumerasi dilakukan akan langsung dites apakah enumerasi ini menghasilkan jawaban yang tepat. Tes dilakukan dengan mengkonversi huruf-huruf yang ada di matriks M dimulai dari kolom paling akhir dengan bantuan list A dan list B kemudian akan dicek apakah hasil penjumlahan pada kolom tersebut benar. Jika ditemukan kolom yang salah tes akan langsung gagal dan dilanjutkan dengan enumerasi berikutnya.
3. Jika tes berhasil, berarti list B berisi nilai-nilai yang tepat untuk karakter pada list A sehingga akan ditampilkan persoalan yang diberikan dan juga solusinya yang didapat dengan mengubah semua karakter pada matriks karakter menjadi angka dengan menggunakan nilai yang terdapat pada list A dan list B. Selain itu ditampilkan juga waktu yang dibutuhkan untuk mendapatkan solusi yang dimulai dari enumerasi pertama sampai penulisan hasil dan juga pada tes seberapa solusi didapat.

Source Program

```
import time

def CtoInt(char, Tab, A):
    #Fungsi untuk mengubah karakter menjadi angka sesuai dengan data yang ada pada matriks Tab dan array A
    #KAMUS LOKAL
    #i : integer
    #ALGORITMA
    if char == "" or char == " ":
        return 0
    else:
        i = 0
        while i < len(Tab[0]) and char != Tab[0][i]:
            i += 1
        return A[i]

def Test(MKata, Tab, First, A):
    #Fungsi untuk melakukan test apakah jawaban yang dicoba sudah benar
    #KAMUS LOKAL
    #success : boolean
    #n, i, total, k : integer
    #ALGORITMA
    success = True
    n = len(MKata)
    i = 0
    total = 0

    for k in First:
        if CtoInt(k, Tab, A) == 0:
            success = False
```

```

while i >= 0 and success:
    for j in range (n-1):
        total += CtoInt(MKata[j][i], Tab, A)
    if total % 10 != CtoInt(MKata[n-1][i], Tab, A):
        success = False
    else:
        total = total // 10
        i -= 1
return success

```

```

def permutasi(n, hasil, First, MKata):

```

#Fungsi untuk membangkitkan permutasi n buah angka (n<= 10) lalu melakukan test

#apakah jawaban sudah sesuai, jika jawaban sudah ditemukan akan merubah variabel global found menjadi True

#ALGORITMA

```

global count

```

```

global found

```

```

global Tab

```

```

if not found:

```

```

    if (n==0):

```

```

        count += 1

```

```

        if Test(MKata, Tab, First, hasil):

```

```

            found = True

```

```

            for k in range (len(Tab[0])):

```

```

                Tab[1][k] = hasil[k]

```

```

    else:

```

```

        i = 0

```

```

        while not found and i <10:

```

```

            if i not in hasil and (i != 0 or (Tab[0][len(hasil)-n] not in First)):

```

```

                hasil[len(hasil)-n] = i

```

```

                permutasi(n-1, hasil, First, MKata)

```

```
        if not found:
            i += 1
        hasil[len(hasil)-n] = -1
```

```
#KAMUS
```

```
#Tab : List of List
```

```
#n,a,b,c,d,e,f,g,h,i,j,k,l, count : integer
```

```
#found : boolean
```

```
#MKata : Matrix of character
```

```
#First, kata, kata1 : List of character
```

```
#ALGORITMA
```

```
Tab = [[],[0 for i in range (10)]]
```

```
n = 0
```

```
#Baca File kemudian disimpan dalam MKata
```

```
file = input("Masukkan nama file: ")
```

```
f = open("../test/"+file, "r")
```

```
MKata = []
```

```
First = []
```

```
kata = f.readline().strip("\n")
```

```
while kata != "":
```

```
    if kata[0] != "-" and kata[0] != "+":
```

```
        if kata.strip(" ")[0] not in First:
```

```
            First.append(kata.strip(" ")[0])
```

```
        kata1 = [" " for i in range (10)]
```

```
        j = 9
```

```
        for i in range(len(kata)-1, -1, -1):
```

```
            kata1[j] = kata[i]
```

```
            j -= 1
```

```

        if kata[i] not in Tab[0] and kata[i] != " ":
            if n < 9:
                Tab[0].append(kata[i])
                n += 1
            else:
                print("Terdapat lebih dari 10 huruf yang berbeda")
    MKata.append(kata1)
    kata = f.readline().strip("\n").strip("+")
f.close()

#Mencatat waktu awal
start = time.time()
count = 0 #jumlah test yang dilakukan
found = False #jawaban sudah sesuai
#Melakukan permutasi kemudian mengetes apakah hasil permutasi merupakan jawaban yang sesuai
#Jika jawaban sudah sesuai loop berhenti

hasil = [-1 for i in range (len(Tab[0]))]
permutasi(len(Tab[0]), hasil, First, MKata)

if found:
    #Mencetak soal
    for k in range (len(MKata)-1):
        print()
        for l in range (10):
            if MKata[k][l] == " ":
                print(" ", end="")
            else:
                print(MKata[k][l], end="")

print("+")

```

```

print(10*" -")
for l in range(10):
    if MKata[len(MKata)-1][l] == " ":
        print(" ", end="")
    else:
        print(MKata[len(MKata)-1][l], end="")
print()
print()

#Mencetak Solusi
print("solusi:")
for k in range (len(MKata)-1):
    print()
    for l in range (10):
        if MKata[k][l] == " " or MKata[k][l] == " ":
            print(" ", end="")
        else:
            print(CtoInt(MKata[k][l], Tab, Tab[1]), end="")

print("+")
print(10*" -")
for l in range(10):
    if MKata[len(MKata)-1][l] == " " or MKata[len(MKata)-1][l] == " ":
        print(" ", end="")
    else:
        print(CtoInt(MKata[len(MKata)-1][l], Tab, Tab[1]), end="")
print()
print()
print("Ditemukan pada tes ke", count)
print("Waktu yang dibutuhkan:", (time.time()-start), "detik")
else:

```

```
    print("Solusi tidak ditemukan")  
input()
```

Hasil *input* dan *output*

No	Input	Output
1	<pre> NUMBER NUMBER+ ----- PUZZLE </pre>	<pre> NUMBER NUMBER+ ----- PUZZLE solusi: 201689 201689+ ----- 403378 Ditemukan pada tes ke 2896579 Waktu yang dibutuhkan: 16.50381326675415 detik </pre>
2	<pre> COCA COLA+ ----- OASIS </pre>	<pre> COCA COLA+ ----- OASIS solusi: 8186 8106+ ----- 16292 Ditemukan pada tes ke 82745 Waktu yang dibutuhkan: 0.2825357913970947 detik </pre>
3	<pre> HERE SHE+ ----- COMES </pre>	<pre> HERE SHE+ ----- COMES solusi: 9454 894+ ----- 10348 Ditemukan pada tes ke 204913 Waktu yang dibutuhkan: 0.9733374118804932 detik </pre>

4	<div> DOUBLE DOUBLE TOIL+ ----- TROUBLE </div>	DOUBLE DOUBLE TOIL+ ----- TROUBLE solusi: 798064 798064 1936+ ----- 1598064 Ditemukan pada tes ke 1375190 Waktu yang dibutuhkan: 8.362718105316162 detik
5	<div> NO GUN NO+ ----- HUNT </div>	NO GUN NO+ ----- HUNT solusi: 87 908 87+ ----- 1082 Ditemukan pada tes ke 83346 Waktu yang dibutuhkan: 0.40221595764160156 detik

6	<div> <div>THREE THREE TWO TWO ONE+</div> <div>THREE THREE TWO TWO ONE+ ----- ELEVEN</div> </div>	<div> <div>THREE THREE TWO TWO ONE+ ----- ELEVEN</div> <div>solusi:</div> <div> 84611 84611 803 803 391+ ----- 171219 </div> <div> Ditemukan pada tes ke 176060 Waktu yang dibutuhkan: 1.2785744667053223 detik </div> </div>
7	<div> <div>CROSS ROADS+</div> <div>CROSS ROADS+ ----- DANGER</div> </div>	<div> <div>CROSS ROADS+ ----- DANGER</div> <div>solusi:</div> <div> 96233 62513+ ----- 158746 </div> <div> Ditemukan pada tes ke 926098 Waktu yang dibutuhkan: 5.093488693237305 detik </div> </div>
8	<div> <div>MEMO FROM+</div> <div>MEMO FROM+ ----- HOMER</div> </div>	<div> <div>MEMO FROM+ ----- HOMER</div> <div>solusi:</div> <div> 8485 7358+ ----- 15843 </div> <div> Ditemukan pada tes ke 63743 Waktu yang dibutuhkan: 0.27347683906555176 detik </div> </div>

Cek List

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)	√	
2. Program berhasil <i>running</i>	√	
3. Program dapat membaca file masukan dan menuliskan luaran.	√	
4. Solusi <i>cryptarithmic</i> hanya benar untuk persoalan <i>cryptarihtmetic</i> dengan dua buah <i>operand</i>		√
5. Solusi <i>cryptarithmic</i> benar untuk persoalan <i>cryptarihtmetic</i> untuk lebih dari dua buah operand.	√	

Drive Kode Program

https://github.com/Saladin21/STIMA_Tucil1