

PENYELESAIAN CRYPTARITHMETIC DENGAN ALGORITMA BRUTE FORCE

LAPORAN TUGAS KECIL 1

Diajukan sebagai laporan dari tugas kecil satu mata kuliah Strategi Algoritma IF2211 pada Semester II Tahun Akademik 2020-2021



Nama : Epata Tuah
NIM : 13519120
Kelas : K-03
Bahasa yang dipilih : Python

**TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021**

A. Algoritma *Brute Force*

1. Ambil semua huruf unik dari gabungan kata-kata operan dan kata hasil penjumlahan operan-operan
2. Simpan huruf awal dari setiap kata pada suatu list. Jika terdapat huruf awal yang sama, elemen list tersebut harus dibuat unik terlebih dahulu
3. Lakukan permutasi n objek berbeda sebanyak r (permutasi nPr) dengan n adalah string angka yang diuji ('0123456789') dan r adalah jumlah elemen dari list huruf awal kata yang unik.
4. Ubah semua hasil permutasi yang awalnya string menjadi array-array berelemen angka
5. Lakukan perulangan untuk setiap array hasil permutasi nPr dalam *range* jumlah hasil permutasi nPr
6. Cek apakah huruf yang sesuai nomor indeks ke- i dari array permutasi adalah huruf awal kata dan angka array indeks ke- i tersebut adalah nol. Jika nol, lewati pemrosesan array tersebut dan lanjutkan ke array hasil permutasi selanjutnya.
7. Jika angka huruf awal tersebut tidak nol, buat list baru yang akan berisi angka dari hasil permutasi yang sudah dipecah sesuai jumlah kata dan indeks kata tersebut (contoh: kata= 'SEND MORE MONEY', list kata unik = [S, E, N, D, M, O, R, Y], salah satu hasil permutasi = [4, 1, 3, 6, 2, 7, 8, 9], list baru = [4136, 2781, 27319])
8. Lakukan penjumlahan elemen list baru indeks ke-0 sampai elemen sebelum elemen indeks terakhir.
9. Cek apakah hasil penjumlahan sama dengan elemen indeks terakhir. Jika sama, cetak hasil luaran dengan ketentuan yang sesuai (angka yang diperoleh, waktu yang dibutuhkan untuk mendapatkan solusi, serta jumlah total tes yang dilakukan untuk mendapatkan solusi), lalu ulangi lagi dari awal dengan membaca array hasil permutasi selanjutnya. Jika tidak sama, cetak luaran tidak dilakukan
10. Ulangi pemrosesan hingga semua array hasil permutasi selesai diproses. Hal ini dilakukan karena solusi dari *cryptarithmic* dapat lebih dari satu.

B. *Source Code*

```
# NIM : 13519120
# Kelas : K-03
# Mata Kuliah : Strategi Algoritma
# Deskripsi : TUGAS KECIL 1 (merancang algoritma Brute Force pada puzzle
Cryptarithmic)

#import library time untuk menghitung waktu yang diperlukan algoritma Brute Force untuk
menemukan solusi
import time

#FUNGSI-FUNGSI TAMBAHAN
#Fungsi mengukur panjang dari string
def panjang(string):
    hitung = 0
    for i in string:
        hitung+= 1
```

```

return(hitung)

#fungsi cacahSatuSatu mencacah array string berdasarkan indeks nya, berupa tuple [(indeks,string),...]
def CacahSatuSatu(string,x=0):
    return [(x+i, string[i]) for i in range(panjang(string))]

#Fungsi kombinasi (digunakan sebagai fungsi bantuan pada fungsi permutasi)
def Kombinasi(string, r):
    if not r:
        yield ""
    elif string:
        for i in Kombinasi(string[1:], r-1):
            yield string[0] + i
        yield from Kombinasi(string[1:], r)

#Fungsi permutasi
def Permutasi(string, r):
    if not r:
        yield ""
    else:
        for hasil in Kombinasi(string, r):
            for i, huruf in CacahSatuSatu(hasil):
                total = hasil[:i] + huruf + hasil[i+1:]
                for j in Permutasi(total, r-1):
                    yield huruf + j

#Fungsi UbahKeList mengubah string menjadi list
def UbahKeList(Kata):
    x = []
    for i in Kata:
        x+=i
    return(x)

#Fungsi GantiKarakter mengganti karakter spesifik pada string
def GantiKarakter(Kata,char1,char2):
    kataBaru = ""
    for i in Kata:
        if (i!=char1):
            kataBaru+=i
        else:
            kataBaru+=char2
    return(kataBaru)

#Fungsi ListHurufKata menjadikan list Kata tersebut terdiri dari
#kata-kata yang unik/berbeda
def ListHurufKata(Kata) :
    x = []
    for i in Kata:

```

```
    if not i in x:
        x+=i
    return(x)
```

#Prosedur UbahListAngka berfungsi mengisi elemen listAngka dengan
#elemen listArray berdasarkan range listKata dan listKataUnik

```
def UbahListAngka(listKata,listKataUnik,listAngka,listArray):
    for i in range(panjang(listKataUnik)):
        for j in range(panjang(listKata)):
            if (listKata[j]==listKataUnik[i]):
                listAngka[j] = listArray[i]
```

def CekBeradaDiIndeksAwal(array,angka,listKataUnik,listHurufAwal):

```
    valid = False
    hitungIndeks = 0
    for i in range(panjang(listKataUnik)):
        for j in range(panjang(listHurufAwal)):
            if (listKataUnik[i]==listHurufAwal[j]):
                hitungIndeks +=1
    listSama = [0 for i in range(hitungIndeks)]
    k = 0
    for i in range(panjang(listKataUnik)):
        for j in range(panjang(listHurufAwal)):
            if (listKataUnik[i]==listHurufAwal[j]):
                listSama[k] = i
                k+=1
    for indeks in listSama:
        if (array[indeks]==angka):
            valid = True
    return(valid)
```

#Fungsi mencari nilai maksimum dari array

def maksimum(array):

```
    if (panjang(array) == 0):
        return 0
```

```
    maks = array[0]
```

```
    for i in array:
        if (i>maks):
            maks = i
```

```
    return(maks)
```

#PROGRAM DIMULAI

#Input awal

```
print("Silahkan cek daftar file teks yang ada di folder directory ../test")
```

```

inputAwal = str(input("Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): "))
fname = "../test/" + inputAwal + ".txt"
#Membuka file
f = open(fname,'r')
mulai = time.process_time() #Waktu dihitung mulai dari line ini
content = f.readlines()
panjangContentAwal = panjang(content)
for i in range(panjangContentAwal):
    content[i] = GantiKarakter(content[i],'\n','') #menganti karakter newline '\n' dengan ''
    print(content[i]) #output isi file
print("")
blankspace = chr(32) #agar spasi (blankspace) dapat diabaikan
old = ['+',blankspace,'-']
new = ['',",",""]
for i in range(panjang(content)):
    for j in range(panjang(old)):
        content[i] = GantiKarakter(content[i],old[j],new[j]) #menghapus karakter yang tidak
        penting
del content[-2] #menghapus content ke-2 terakhir (hanya berisi array "", agar tidak redundant)

#menyatukan semua kata yang ada di array content
#contoh:['SEND','MORE','MONEY'] menjadi 'SENDMOREMONEY'
Kata = content[0]
for i in range(1,panjang(content)):
    Kata = Kata+content[i]

#Mengisi huruf awal setiap kata pada array listHurufAwal
listHurufAwal = ['*' for i in range(panjang(content))]
for i in range(panjang(content)):
    listHurufAwal[i] = UbahKeList(content[i])[0]
listHurufAwal = ListHurufKata(listHurufAwal) #menjadikan isi array unik

#Mencari length kata terpanjang
listPanjang = [0 for i in range(panjang(content))]
for i in range(panjang(content)):
    listPanjang[i] = panjang(content[i])
kataTerpanjang = maksimum(listPanjang)

#INISIALISASI LIST
#list kata lengkap, contoh: ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'E', 'M', 'O', 'N', 'E', 'Y']
listKata = UbahKeList(Kata)

#list kata unik, contoh: ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y']
listKataUnik = ListHurufKata(listKata)

#list angka kosong, akan diisi dengan list angka permutasi
#contoh: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
listAngka = [0 for i in range(panjang(UbahKeList(Kata)))]

```

```

#list indeks, digunakan untuk indeks listAngka yang akan disesuaikan
#dengan operand ke berapa atau hasil
#contoh: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
listIndeks = [0 for i in range(panjang(UbahKeList(Kata)))]
for i in range(panjang(listIndeks)):
    listIndeks[i] = i

#ALGORITMA DIMULAI
nilaiPermutasi = Permutasi('0123456789',panjang(listKataUnik))
hitungPercobaan = 0
jumlahSolusi = 0
for angka in nilaiPermutasi:
    hitungPercobaan+=1
    array = [int(x) for x in angka]
    #Cek apakah terdapat huruf awal kata bernilai nol. Jika ada skip ke array permutasi
    selanjutnya
    if (CekBeradaDiIndeksAwal(array,0,listKataUnik,listHurufAwal)==True):
        continue
    else:
        #mengubah listAngka dengan elemen array sesuai range listKata dan listKataUnik
        UbahListAngka(listKata,listKataUnik,listAngka,array)
        operand = [" for i in range(panjang(content))"] #inisialisasi awal array operand
        count = 0
        #menghasilkan operand-operand sebanyak isi array content
        for i in range(panjang(content)):
            for j in range(panjang(content[i])):
                if (i==0):
                    operand[i]+= str(listAngka[j])
                else:
                    operand[i]+=str(listAngka[j+count])
                count+=panjang(content[i])
        #menjumlahkan operand-operand
        hasil = int(operand[0])
        for i in range(1,panjang(content)-1):
            hasil+= int(operand[i])
        #jika angka terakhir tidak sesuai dengan penjumlahan operand-operandnya, skip ke
        array permutasi selanjutnya
        if (int(operand[panjang(content)-1])!=hasil):
            continue
        #jika hasil sesuai dengan penjumlahan operand-operandnya, cetak luaran sesuai
        ketentuan
        else:
            jumlahSolusi+=1
            for i in range(panjang(content)-1):
                if (i==panjang(content)-2):
                    print(str(f"{int(operand[i]):>{kataTerpanjang}}")+'+')
                else:

```

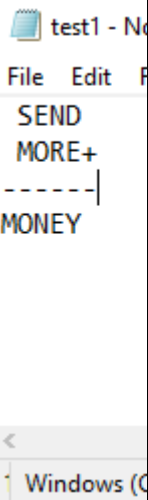
```


        print(str(f"{int(operand[i]):>{kataTerpanjang}}"))
    print('-----')
    print(str(f"{int(operand[panjang(content)-1]):>{kataTerpanjang}}"),'\n')
    if (jumlahSolusi>1):
        print("Waktu eksekusi saat menemukan solusi ke-",jumlahSolusi," :
",time.process_time() - mulai," detik")
        print("Jumlah total tes yang dilakukan pada solusi ke-
",jumlahSolusi," : ",hitungPercobaan,'\n')
    else:
        print("Waktu eksekusi saat menemukan solusi: ",time.process_time()
- mulai," detik")
        print("Jumlah total tes yang dilakukan: ",hitungPercobaan,'\n')
    print("Jumlah solusi yang ditemukan:", jumlahSolusi)
    print("Waktu eksekusi hingga program berakhir: ",time.process_time()-mulai," detik")

```

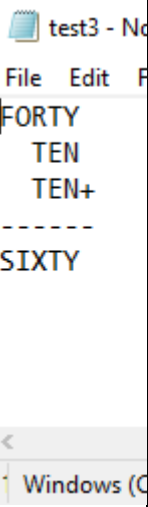
C. Hasil Masukan dan Luaran

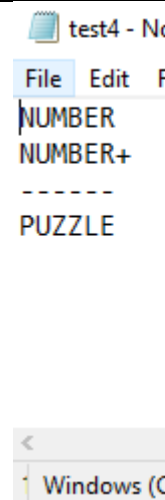
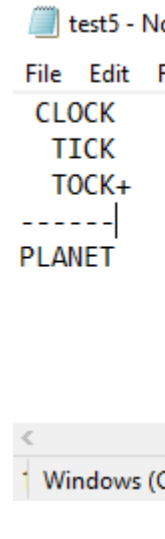
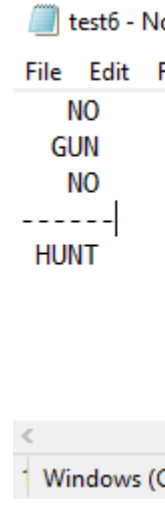
Keterangan: Kode dites pada python versi 3.9 64-bit, OS Windows 10, prosesor Intel Core i3-4030u, RAM 8 GB, serta storage berupa HDD.

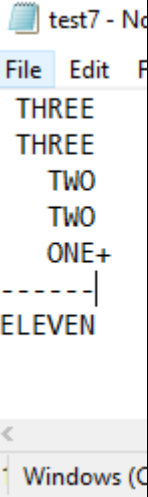
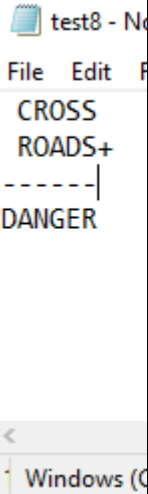
No.	Masukan	Luaran
1.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test</p> <p>Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test1</p> <p>SEND</p> <p>MORE+</p> <p>-----</p> <p>MONEY</p> <p>9567</p> <p>1085+</p> <p>-----</p> <p>10652</p> <p>Waktu eksekusi saat menemukan solusi: 67.21875 detik</p> <p>Jumlah total tes yang dilakukan: 844280</p> <p>Jumlah solusi yang ditemukan: 1</p> <p>Waktu eksekusi hingga program berakhir: 149.6875 detik</p> <p>>>> </p>

2.	 test2 - Notepad File Edit Format View Help JUNE JULY+ ----- APRIL	<p>Silahkan cek daftar file teks yang ada di folder directory ../test</p> <p>Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test2</p> <pre> JUNE JULY+ ----- APRIL 5486 5437+ ----- 10923 Waktu eksekusi saat menemukan solusi: 179.8125 detik Jumlah total tes yang dilakukan: 2009240 5487 5436+ ----- 10923 Waktu eksekusi saat menemukan solusi ke- 2 : 179.875 detik Jumlah total tes yang dilakukan pada solusi ke- 2 : 2009960 7924 7906+ ----- 15830 Waktu eksekusi saat menemukan solusi ke- 3 : 266.375 detik Jumlah total tes yang dilakukan pada solusi ke- 3 : 2875036 7926 7904+ ----- 15830 </pre>
----	---	---

	<p>Waktu eksekusi saat menemukan solusi ke- 4 : 266.5 detik Jumlah total tes yang dilakukan pada solusi ke- 4 : 2876452</p> <p>8432 8475+ ----- 16907</p> <p>Waktu eksekusi saat menemukan solusi ke- 5 : 287.125 detik Jumlah total tes yang dilakukan pada solusi ke- 5 : 3081418</p> <p>8435 8472+ ----- 16907</p> <p>Waktu eksekusi saat menemukan solusi ke- 6 : 287.1875 detik Jumlah total tes yang dilakukan pada solusi ke- 6 : 3082138</p> <p>8524 8539+ ----- 17063</p> <p>Waktu eksekusi saat menemukan solusi ke- 7 : 290.78125 detik Jumlah total tes yang dilakukan pada solusi ke- 7 : 3117227</p> <p>8529 8534+ ----- 17063</p>
--	--

		<p>Waktu eksekusi saat menemukan solusi ke- 8 : 290.96875 detik Jumlah total tes yang dilakukan pada solusi ke- 8 : 3119339</p> <pre> 8532 8564+ ----- 17096 </pre> <p>Waktu eksekusi saat menemukan solusi ke- 9 : 291.203125 detik Jumlah total tes yang dilakukan pada solusi ke- 9 : 3121617</p> <pre> 8534 8562+ ----- 17096 </pre> <p>Waktu eksekusi saat menemukan solusi ke- 10 : 291.28125 detik Jumlah total tes yang dilakukan pada solusi ke- 10 : 3122337</p> <pre> 9253 9207+ ----- 18460 </pre> <p>Waktu eksekusi saat menemukan solusi ke- 11 : 315.84375 detik Jumlah total tes yang dilakukan pada solusi ke- 11 : 3368237</p> <pre> 9257 9203+ ----- 18460 </pre> <p>Waktu eksekusi saat menemukan solusi ke- 12 : 316.0625 detik Jumlah total tes yang dilakukan pada solusi ke- 12 : 3370349</p> <p>Jumlah solusi yang ditemukan: 12 Waktu eksekusi hingga program berakhir: 341.71875 detik >>></p>
3.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test3</p> <pre> FORTY TEN TEN+ ----- SIXTY 29786 850 850+ ----- 31486 </pre> <p>Waktu eksekusi saat menemukan solusi: 95.390625 detik Jumlah total tes yang dilakukan: 1083579</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 362.5 detik >>> </p>

4.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test4</p> <p>NUMBER NUMBER+ ----- PUZZLE</p> <p>201689 201689+ ----- 403378</p> <p>Waktu eksekusi saat menemukan solusi: 147.671875 detik Jumlah total tes yang dilakukan: 1532493</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 359.90625 detik</p> <p>>>> </p>
5.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test5</p> <p>CLOCK TICK TOCK+ ----- PLANET</p> <p>90892 6592 6892+ ----- 104376</p> <p>Waktu eksekusi saat menemukan solusi: 350.953125 detik Jumlah total tes yang dilakukan: 3302475</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 387.015625 detik</p> <p>>>> </p>
6.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test6</p> <p>NO GUN NO ----- HUNT</p> <p>87 908 87+ ----- 1082</p> <p>Waktu eksekusi saat menemukan solusi: 1.515625 detik Jumlah total tes yang dilakukan: 25051</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 10.421875 detik</p> <p>>>> </p>

7.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test7</p> <pre> THREE THREE TWO TWO ONE+ ----- ELEVEN 84611 84611 803 803 391+ ----- 171219 </pre> <p>Waktu eksekusi saat menemukan solusi: 200.796875 detik Jumlah total tes yang dilakukan: 1756932</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 428.453125 detik >>> </p>
8.		<p>Silahkan cek daftar file teks yang ada di folder directory ../test Ketik nama file yang ingin dites (misal 'test1' tanpa tanda petik): test8</p> <pre> CROSS ROADS+ ----- DANGER 96233 62513+ ----- 158746 </pre> <p>Waktu eksekusi saat menemukan solusi: 337.140625 detik Jumlah total tes yang dilakukan: 3614574</p> <p>Jumlah solusi yang ditemukan: 1 Waktu eksekusi hingga program berakhir: 338.765625 detik >>> </p>

D. Alamat drive

Link Repository: <https://github.com/epata/Tucil-1-Strategi-Algoritma---13519120.git>

Link Alternatif: <https://drive.google.com/drive/u/0/folders/1K8Je2A0MtOPB7cFfovHMxX-fpC1uHGYf>

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (<i>no syntax error</i>)	✓	
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>cryptarithmic</i> dengan dua buah <i>operand</i>		✓

5. Solusi <i>cryptarithmic</i> benar untuk persoalan <i>cryptarithmic</i> untuk lebih dari dua buah operand.	✓	
--	---	--