

# **LAPORAN TUGAS KECIL 1**

## **STRATEGI ALGORITMA - IF2211**

**“Penyelesaian Cryptarithmic dengan Algoritma *Brute Force*”**



**Dibuat Oleh :**

**Mohammad Sheva Almeyda Sofjan - 13519018**

**K-01**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2021**

## DESKRIPSI PERSOALAN

*Cryptarithmic* (atau *cryptarithm*) adalah sebuah *puzzle* penjumlahan di dalam matematika dimana angka diganti dengan huruf. Setiap angka direpresentasikan dengan huruf yang berbeda. Deskripsi permainan ini adalah: diberikan sebuah penjumlahan huruf, carilah angka yang merepresentasikan huruf-huruf tersebut.

Sebagai contoh, persoalan berikut :

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Memiliki solusi :

$$\begin{array}{r} 9567 \\ + 1085 \\ \hline 10652 \end{array}$$

Dengan  $S = 9$ ,  $E = 5$ ,  $N = 6$ ,  $D = 7$ ,  $M = 1$ ,  $O = 0$ ,  $R = 8$ , dan  $Y = 2$

Akan dibuat program yang mampu menyelesaikan persoalan penjumlahan cryptarithmic menggunakan algoritma *brute force*.

## ALGORITMA BRUTE FORCE

Secara umum, gambaran langkah-langkah program dalam menemukan solusi dari input persoalan adalah sebagai berikut:

1. Program menjalankan file main.py, dan mulai mengeksekusi fungsi *reader()* yang terdapat pada util.py untuk membaca input program, melakukan validasi terhadap input dan mengembalikan 2 list yaitu list yang berisi string operand (list isi) dan hasil dan list yang berisi karakter unik pada persoalan (list charlist).
2. Mengisi list charlist hingga elemen yang terdapat pada list charlist berjumlah 10 (jumlah elemen maksimum), bila banyak karakter unik kurang dari 10 maka list akan diisi dengan karakter “\*” hingga banyak elemen pada list menjadi 10 buah.
3. Melakukan permutasi pada list charlist yang sudah diubah sedemikian rupa pada langkah (2), dengan rincian :
  - i. Buat list kosong, beri nama permlist
  - ii. Kasus basis : jika panjang list charlist = 1, maka fungsi akan mereturn list charlist
  - iii. Kasus rekurens : Jika panjang list charlist > 1, maka fungsi akan melakukan iterasi dari elemen pertama hingga elemen terakhir charlist untuk mencari permutasi dari elemen-elemen pada charlist.  
Pencarian permutasi elemen dilakukan dengan menjadikan elemen ke-i (i merupakan indeks charlist pada iterasi) sebagai elemen pertama relatif pada suatu list permutasi yang baru, kemudian dicari permutasi elemen-elemen sisa (selain elemen ke-i yang tadi) dengan proses yang sama, dan akhirnya append (konkatenasi) antara elemen ke-i yang tadi (sebagai elemen pertama relatif) dengan permutasi elemen sisanya. Lakukan hingga elemen indeks terakhir pada list charlist menjadi elemen pertama pada list permutasi permlist.
4. Filter hasil permutasi yang dilakukan pada langkah (3), untuk menghilangkan duplikat hasil permutasi yang diakibatkan oleh penambahan karakter “\*” lebih dari satu kali (untuk persoalan dengan banyak karakter unik kurang dari 9 karakter).
5. Untuk setiap hasil permutasi yang sudah di filter, lakukan hal berikut (Implementasi Brute Force) :
  - i. Buat list yang berisi value dari suatu string pada list isi, list ini akan dinamakan cvlist (char value list). Value untuk setiap karakter mengacu

pada indeks karakter tersebut dalam list yang di *generate* pada permutasi (list charlist yang sudah difilter).

- ii. Pada char value list, cek apakah huruf pertama pada list isi memiliki value 0 atau tidak, jika value nya bukan 0 maka lanjut ke langkah selanjutnya
- iii. Pada char value list, cek apakah sum dari semua operand sama dengan hasil (cek sum elemen 0 hingga sebelum terakhir sama dengan value elemen terakhir pada char value list), jika sama maka proses pencarian solusi di terminasi dan mulai menuliskan output, sehingga solusi yang di output terbatas kepada satu solusi saja.

Sebagai contoh, apabila program diberikan input persoalan SEND + MORE = MONEY (seperti pada bagian deskripsi persoalan), maka program akan menjalankan hal-hal berikut :

1. Program membaca persoalan dan hasilnya adalah :  
 List isi = ['SEND', 'MORE', 'MONEY']  
 List charlist = ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y']
2. Dikarenakan banyak karakter dalam list charlist kurang dari 10, maka list charlist akan diisi dengan karakter "\*" sebanyak 2 buah sehingga :  
 List charlist = ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y', '\*', '\*']
3. Akan dilakukan permutasi terhadap list charlist, dengan rincian sebagai berikut :  
 Untuk iterasi pertama, elemen dengan indeks pertama pada list charlist (elemen 'S') akan menjadi elemen pertama pada *sublist* pada list permlist, lalu dicari permutasi elemen sisanya. Sehingga untuk iterasi pertama ( i = 0, elemen 'S' sebagai first) didapat beberapa hasil permutasi pertama sebagai berikut :  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y', '\*', '\*']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y', '\*', '\*']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', '\*', 'Y', '\*']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', '\*', '\*', 'Y']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', '\*', 'Y', '\*']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'R', '\*', '\*', 'Y']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'Y', 'R', '\*', '\*']  
 ['S', 'E', 'N', 'D', 'M', 'O', 'Y', 'R', '\*', '\*']  
 .....  
 .....

Elemen elemen akhir list terlihat seperti di tukar / *swap* satu persatu. Perhatikan bahwa terdapat beberapa hasil permutasi yang sama dikarenakan terdapat lebih dari satu karakter '\*' dan karakter tersebut dianggap *distinct*.

4. Dikarenakan terdapat beberapa hasil permutasi yang berulang, maka hasil yang berulang akan difilter dan dihilangkan sehingga hasil untuk contoh pada nomor 3 menjadi :

```
['S', 'E', 'N', 'D', 'M', 'O', 'R', 'Y', '*', '*']
['S', 'E', 'N', 'D', 'M', 'O', 'R', '*', 'Y', '*']
['S', 'E', 'N', 'D', 'M', 'O', 'R', '*', '*', 'Y']
['S', 'E', 'N', 'D', 'M', 'O', 'R', '*', 'Y', '*']
['S', 'E', 'N', 'D', 'M', 'O', 'Y', 'R', '*', '*']
```

.....  
 .....

5. Assign value pada list isi sesuai hasil permutasi dan lakukan proses validasi.

Sublist permutasi yang memenuhi persoalan adalah :

Sublist permlist = ['O', 'M', 'Y', '\*', '\*', 'E', 'N', 'D', 'R', 'S']

List isi (awal) = ['SEND', 'MORE', 'MONEY']

List cvlist (char value list) = ['9567', '1085', '10652']

Sehingga didapat solusi dari persoalan yang diberikan.

## SOURCE CODE PROGRAM

Program dibuat menggunakan bahasa Python (Python 3). Berikut adalah *source code* dari program yang telah dibuat.

main.py

```
'''
Nama      : Mohammad Sheva Almeyda Sofjan
NIM/Kelas : 13519018/K01
Deskripsi : Tugas Kecil 1 IF2211 Strategi Algoritma
            Cryptarithmic Solver menggunakan Algoritma Brute Force
            (Only supports additions)
Deskripsi File : File Utama (Driver)
'''
import sys # Buat auto exit program
```

```

from time import time # Library Waktu
from util import reader,outputter # Utility
from sols import permute,slotfill,charvalue,sumChecker,isZero,permForm #
Algo Solusi

print("\nCrytarithmetic Solver")
print("Dapat menyelesaikan persoalan penjumlahan cryptarithmic")
print("Cr : Mohammad Sheva (13519018)\n")

parse = reader() # Parsing dari text file
inittime = time() # Initial Time

print("\nInput anda :\n")
outputter(parse[1])
print("Silakan tunggu, proses ini mungkin agak lama.\n")
if (len(parse[1][len(parse[1])-1]) < len(parse[1][0])): # Cek apakah
    banyak digit hasil lebih kecil atau lebih besar dari operand
    print("Tidak ditemukan solusi dari input anda")
    print("Lama eksekusi : ",time()-inittime," detik")
    sys.exit()

count = 0 # Counter jumlah tes
havesol = False # Apakah persoalan ini dapat ditemukan solusinya?
perm = permForm(permute(slotfill(parse[2]))) # Generate List of
Permutation
for data in perm: # Iterasi untuk cek assigned value dari tiap char pada
tiap permutasi
    cvlist=charvalue(parse[1],data) # Assign value ke char
    count+=1
    if(not isZero(cvlist)): # Cek leading zero pada operand/hasil
        if (sumChecker(cvlist)): # Cek kesamaan sum operand dan hasil
            havesol = True
            break # Terbatas ke 1 solusi

if(havesol):
    print("Solusi dari input anda :\n")
    outputter(cvlist)
else:
    print("Tidak ditemukan solusi dari input anda")

print("Total tes : ",count)

```

```
print("Lama eksekusi : ",time()-inittime," detik"
```

sols.py

```
'''
Nama      : Mohammad Sheva Almeyda Sofjan
NIM/Kelas : 13519018/K01
Deskripsi : Tugas Kecil 1 IF2211 Strategi Algoritma
            Cryptarithmic Solver menggunakan Algoritma Brute Force
            (Only supports additions)
Deskripsi File : Algoritma Penyelesaian
'''
def slotfill(charlist):
    '''
    Mengisi slot pada list of unique char hingga maksimum (pada kasus
    ini,10) dengan dummy character "*"
    charlist: List yang berisi karakter unik dari operand dan hasil
    '''
    slotlist = charlist
    while(len(slotlist)<10):
        slotlist.append("*")
    return slotlist

def permute(charlist):
    '''
    Mencari permutasi dari list of unique char (charlist);naive,
    rekursif
    '''
    permlist = [] #List permutasi dari string charlist
    if len(charlist)==0: #Kasus List Kosong
        return []
    elif len(charlist)==1: # Basis List 1 karakter
        return charlist
    else:
        # Rekurens; melakukan iterasi kepada tiap karakter dalam list
        sehingga terbentuk list permutasi
        # dengan karakter tersebut sebagai elemen pertama, dilakukan
```

```

recursively
    for i in range(len(charlist)):
        first = charlist[i] # Ambil elemen ke i sebagai index
        pertama dari suatu list permutasi
        rest = charlist[:i] + charlist[i+1:] # Elemen sisa
        for p in permute(rest): #Iterasi permutasi elemen sisa
            secara rekursif
                # Menambahkan elemen ke i sebagai index 0 dalam list of
            list permutasi,
                # dilanjutkan dengan menambahkan permutasi selanjutnya
            ke permlist secara rekursif
                # dengan memanggil kembali fungsi permute
                permlist.append(first+p)

    return permlist

def charvalue(isi,charlist):
    '''
    Assign value untuk tiap operand dan hasil
    isi : List yang berisi operand dan hasil
    charlist: List yang berisi karakter unik dari operand dan hasil
    '''
    cvlist = []
    for kalimat in isi:
        opvalue = ""
        for char in kalimat:
            opvalue+=str(charlist.index(char))
        cvlist.append(opvalue)
    return cvlist

def sumChecker(cvlist):
    '''
    Buat cek kesamaan sum operand dengan hasil
    cvlist : List yang berisi value dari tiap operand dan hasil
    '''
    opsum = 0
    for i in range(len(cvlist)-1):
        opsum+=int(cvlist[i])
    hasil = int(cvlist[len(cvlist)-1])
    return (opsum==hasil)

```



```

def isZero(cvlist):
    '''
    Cek char pertama dalam operand atau hasil bernilai nol atau tidak
    cvlist : List yang berisi value dari tiap operand dan hasil
    '''
    iszero = False
    for cv in cvlist:
        if int(cv[0])==0:
            iszero = True
            break
    return iszero

def permForm(charlist):
    '''
    Buat ubah format hasil permutasi
    '''
    charlist = list(dict.fromkeys(charlist)) # Menghapus duplikat
    permutasi
    rdlist = []
    for p in charlist:
        rdlist.append(list(p))
    return rdlist

```

util.py

```

'''
Deskripsi File : Fungsi Utility
'''
'''
Nama      : Mohammad Sheva Almeyda Sofjan
NIM/Kelas : 13519018/K01
Deskripsi : Tugas Kecil 1 IF2211 Strategi Algoritma
            Cryptarithmic Solver menggunakan Algoritma Brute Force
            (Hanya support penjumlahan)
'''

def reader():
    '''
    Reader file text, mereturn nama file, list isi file perbaris, dan

```

```

list char unik
'''
notfound = True
while notfound :
    fname = input("Masukkan Nama File (dengan ekstensi, sebagai
contoh : tcl.txt) : ")
    try:
        file = open('../test/'+fname,"r")
        notfound = False
    except:
        print("File tidak ditemukan. Ulangi lagi")

isi = file.readlines() # Membaca isi file
file.close()
# Mengubah ke array, menghapus karakter tak guna
for i in range(len(isi)):
    isi[i] = isi[i].replace("\n","")
    isi[i] = isi[i].replace(" ", "")

# Format Checking
if((len(isi) < 4) or (isi[len(isi)-2]!="-----") or
(isi[len(isi)-3][len(isi)-3]-1]!="+")):
    print("File tidak sesuai format, pembacaan file akan
diulangi...")
    return reader()

isi[len(isi)-3]=isi[len(isi)-3][:-1] #Remove plus sign di operand
terakhir
isi[len(isi)-2] = isi[len(isi)-1] #Delete -----
isi = isi[:-1]

# Another format checking dan penambahan list berisi karakter unik
pada operand/hasil
charlist = []
isalfa = True
for string in isi:
    if(not string.isalpha()):
        isalfa = False
        break
for kar in string : #List of unique chars
    try:

```

```

        charlist.index(kar)
    except:
        charlist.append(kar)

    if((not isalfa) or (len(charlist)>10)):
        print("File tidak sesuai format, pembacaan file akan
diulangi...")
        return reader()

    print("Pembacaan file berhasil")

    return fname,isi,charlist

def outputter(list):
    '''
    Mengoutput persoalan maupun solusi sesuai format
    '''
    for i in range(len(list)-2): #Iterate hingga sebelum baris yang ada
plus sign nya
        space = len(list[len(list)-1]) - len(list[i])
        print(space*' '+list[i])

    space = len(list[len(list)-1]) - len(list[len(list)-2]) # Baris yang
ada plus sign nya
    print(space*' '+list[len(list)-2]+"+")
    print("-----")
    print(list[len(list)-1],"\n")

```

## SCREENSHOT HASIL TESTING

### - Test Case 1

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc1.txt
Pembacaan file berhasil

Input anda :

NUMBER
NUMBER+
-----
PUZZLE

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

201689
201689+
-----
403378

Total tes : 406541
Lama eksekusi : 14.471103429794312 detik
```

### - Test Case 2

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc2.txt
Pembacaan file berhasil

Input anda :

TILES
PUZZLES+
-----
PICTURE

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

91542
3077542+
-----
3169084

Total tes : 2235160
Lama eksekusi : 27.868641138076782 detik
```

- Test Case 3

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc3.txt
Pembacaan file berhasil

Input anda :

CLOCK
TICK
TOCK+
-----
PLANET

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

90892
6592
6892+
-----
104376

Total tes : 578730
Lama eksekusi : 16.746838092803955 detik
```

- Test Case 4

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc4.txt
Pembacaan file berhasil

Input anda :

COCA
COLA+
-----
OASIS

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

8186
8106+
-----
16292

Total tes : 47657
Lama eksekusi : 7.912884473800659 detik
```

- Test Case 5

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc5.txt
Pembacaan file berhasil

Input anda :

    HERE
    SHE+
    -----
    COMES

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

    9454
    894+
    -----
    10348

Total tes : 335004
Lama eksekusi : 9.873001337051392 detik
```

- Test Case 6

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc6.txt
Pembacaan file berhasil

Input anda :

    DOUBLE
    DOUBLE
    TOIL+
    -----
    TROUBLE

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

    798064
    798064
    1936+
    -----
    1598064

Total tes : 1329716
Lama eksekusi : 24.515043258666992 detik
```

- Test Case 7

```

Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc7.txt
Pembacaan file berhasil

Input anda :

    NO
    GUN
    NO+
-----
HUNT

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

    87
    908
    87+
-----
1082

Total tes : 51237
Lama eksekusi : 7.8310651779174805 detik

```

- Test Case 8

```

Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc8.txt
Pembacaan file berhasil

Input anda :

    THREE
    THREE
    TWO
    TWO
    ONE+
-----
ELEVEN

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

    84611
    84611
    803
    803
    391+
-----
171219

Total tes : 1605107
Lama eksekusi : 29.29597568511963 detik

```

- Test Case 9

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc9.txt
Pembacaan file berhasil

Input anda :

CROSS
ROADS+
-----
DANGER

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

96233
62513+
-----
158746

Total tes : 3479700
Lama eksekusi : 37.15782856941223 detik
```

- Test Case 10

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : tc10.txt
Pembacaan file berhasil

Input anda :

MEMO
FROM+
-----
HOMER

Silakan tunggu, proses ini mungkin agak lama.

Solusi dari input anda :

8485
7358+
-----
15843

Total tes : 130041
Lama eksekusi : 8.482404232025146 detik
```



- Nosol.txt (tidak memiliki solusi)

```
Masukkan Nama File (dengan ekstensi, sebagai contoh : tc1.txt) : nosol.txt
Pembacaan file berhasil

Input anda :

SATU
SATU+
-----
DUA

Silakan tunggu, proses ini mungkin agak lama.

Tidak ditemukan solusi dari input anda
Lama eksekusi : 0.0 detik
```

## TAUTAN SOURCE CODE

Github : <https://github.com/moshval/cryptarithm-solver>

## TABEL PENILAIAN

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> <b>hanya benar</b> untuk persoalan <i>cryptarithmic</i> dengan dua buah <i>operand</i>		✓
5. Solusi <i>cryptarithmic</i> <b>benar</b> untuk persoalan	✓	

<i>cryptarithmic</i> untuk <b>lebih dari</b> dua buah <i>operand</i>		
--	--	--