

Laporan
Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II Tahun 2020/2021

Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*

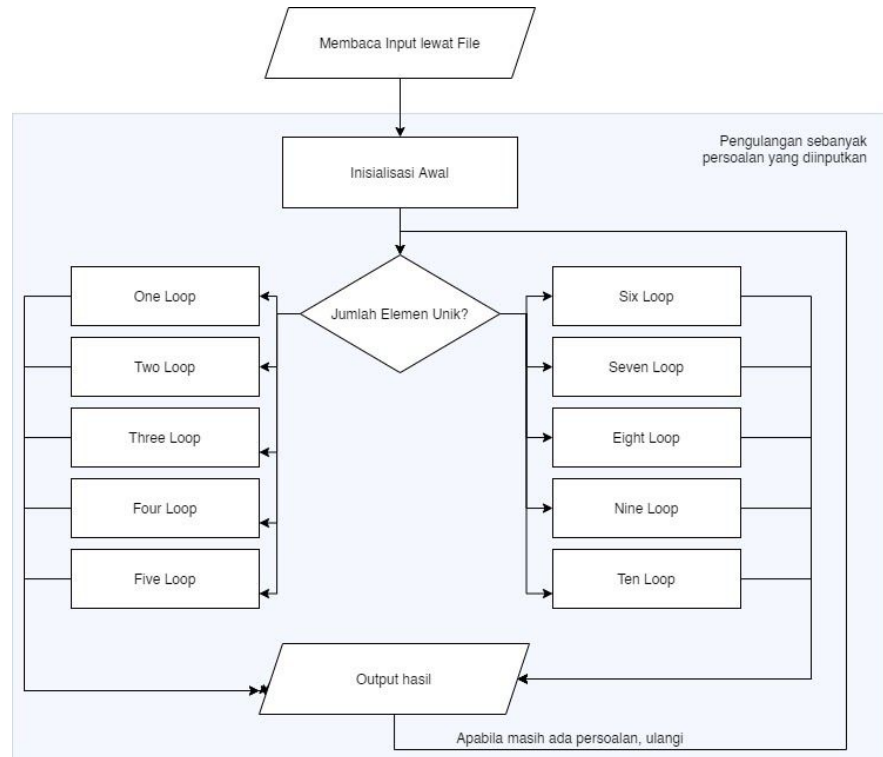


Shaffira Alya Mevia
13519083

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2021

1. Algoritma *Brute Force* dalam Persoalan

Pada persoalan tugas kecil ini, untuk menyelesaikan masalah *cryptarithmic* memerlukan beberapa langkah yang apabila di visualisasikan dengan *flowchart* adalah sebagai berikut.



Gambar 1.1 *Flowchart* sederhana untuk menggambarkan cara kerja program

Berikut adalah penjelasan langkah-langkah algoritma *brute force* yang digunakan secara lebih detail.

- 1) Membuat fungsi untuk inisialisasi data. Inisialisasi data yang digunakan untuk memudahkan proses menggunakan tipe data *dictionary* yang berisikan sebagai berikut.
 - Array huruf unik pada persoalan
 - Array kosong dengan ukuran yang sama dengan array huruf unik untuk melakukan *assigning*
 - Array berisi huruf yang muncul diawal kata dalam bentuk index yang bereferensi kepada array unik
 - Array solusi-solusi yang berhasil
 - Array persoalan
 - Variabel waktu
 - Variabel jumlah iterasi
- 2) Membuat fungsi untuk mengecek kemungkinan solusi. Pada fungsi ini dilakukan pengecekan dengan mengganti huruf pada persoalan dengan angka yang

sesuai. Apabila hasilnya sesuai, maka solusi tersebut akan dimasukkan ke dalam array solusi-solusi yang berhasil.

- 3) Membuat fungsi untuk permutasi. Pada fungsi yang ini konsepnya sangatlah sederhana karena hanya memanfaatkan *konsep nested loop*. Dibatasi sepuluh fungsi permutasi yang berbeda sesuai dengan jumlah huruf yang unik pada suatu persoalan.
- 4) Menjalankan *driver* program pada file utama. Program akan mengiterasikan permasalahan yang dimasukkan oleh pengguna kemudian akan melakukan inisialisasi awal dan menentukan jenis permutasi yang mana yang digunakan berdasarkan jumlah huruf yang unik.

2. Source Code Program

Untuk repository dari program ini dapat diakses melalui [GitHub](#). Sementara itu berikut adalah *source code* untuk main.py.

```
import time
import loops # File fungsi iterasi

print("==== CRYPTARITHMETIC SOLVER ====")
print("This program is using a brute force algorithm, so it may take some time depending on your computer power.")
print("You can add more than 1 (one) problem each file. Use `./folder_name/file_name` if it's on another directory.")
print("")

#----- INPUT PROBLEMS -----#
# Read File
filename = input("Enter filename (.txt only): ")
f = open(filename+".txt", "r")

# Assign to Array
i = 0; j = 0; strAll = []; strTemp = []
for x in f:
    if (x == '\n'):
        x = f.readline()
        strAll.append(strTemp); strTemp = []

    if (x[0] == '-'):
        x = f.readline()
```

```

        x = x.strip()
        strTemp.append(x)
    else:
        x = x.strip().strip('+')
        strTemp.append(x)
strAll.append(strTemp)

#----- ASSIGN AND TIMER -----#
# Timer On
start = time.time()

# Assign to Unique Array
def initCryp(arr):
    # Find Unique
    temp = []; res = []
    for i in arr:
        if (not(i[0] in temp)): temp.append(i[0])
        for j in i:
            #if (not([j, 0] in res) and not(j.isspace())):
            if (not(j in res) and not(j.isspace())):
                res.append(j)

    arrFirst = []
    for i in temp:
        arrFirst.append(res.index(i))

    # Init Dict
    d = {'unique' : res, 'assign' : [0 for i in range(len(res))],
        'firstPos' : arrFirst, 'combi' : [],
        'arr' : arr, 'time' : 0, 'iter' : 0}

    return d

#----- SOLVER -----#
# Permutate
for problem in strAll:
    print('--- Solving Problem ---')
    s = ' + '.join(problem[:len(problem)-1])
    s = s + ' = ' + problem[len(problem)-1]

```

```

print("Problem: ", s)

d = initCryp(problem)
length = len(d['unique'])
if (length == 1):
    loops.one_loop(d)
elif (length == 2):
    loops.two_loop(d)
elif (length == 3):
    loops.three_loop(d)
elif (length == 4):
    loops.four_loop(d)
elif (length == 5):
    loops.five_loop(d)
elif (length == 6):
    loops.six_loop(d)
elif (length == 7):
    loops.seven_loop(d)
elif (length == 8):
    loops.eight_loop(d)
elif (length == 9):
    loops.nine_loop(d)
elif (length == 10):
    loops.ten_loop(d)
else:
    print("Error.")

# Results
print("")
print('--- Results ---')
print(len(d['combi']), " solution(s) found." )
for i in range(len(d['unique'])):
    for j in range(len(d['combi'])):
        print(d['unique'][i], "=", d['combi'][j][i], " ", end="")
    print('')

print("Iterations: ", d['iter'])

d['time'] = time.time() - start

```

```

print("Time taken: ", d['time'])
#print(d)

#----- DELAY -----#
input('Press ENTER to exit. To try again, you need to close and reopen the
program.')
```

3. Hasil Program

Berikut adalah beberapa *screenshot* dan persoalan yang digunakan dalam *testing* program.

1) SEND + MORE = MONEY

Masukan dari program berupa file yang berisi seperti demikian.

```

SEND
MORE+
-----
MONEY
```

Kemudian hasil yang dikeluarkan oleh program adalah sebagai berikut.

```

===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): ../test/test1
--- Solving Problem ---
Problem: SEND + MORE = MONEY

--- Results ---
1 solution(s) found.
S = 9
E = 5
N = 6
D = 7
M = 1
O = 0
R = 8
Y = 2
Iterations: 1451520
Time taken: 148.74766421318054
Press ENTER to exit. To try again, you need to close and reopen the program.
```

Gambar 3.1 Hasil keluaran dari program.

2) MEMO + FROM = HOMER

Untuk masukkannya, memiliki format yang sama seperti uji kasus 1.

Hasilnya adalah sebagai berikut.

```
Cryptarithmic Solver

===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test9
--- Solving Problem ---
Problem: MEMO + FROM = HOMER

--- Results ---
1 solution(s) found.
M = 8
E = 4
O = 5
F = 7
R = 3
H = 1
Iterations: 105840
Time taken: 2.486879825592041
Press ENTER to exit. To try again, you need to close and reopen the program.
```

Gambar 3.2 Hasil keluaran dari program.

3) COCA + COLA = OASIS

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1.
Hasilnya adalah sebagai berikut.

```
Cryptarithmic Solver

===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test3
--- Solving Problem ---
Problem: COCA + COLA = OASIS

--- Results ---
1 solution(s) found.
C = 8
O = 1
A = 6
L = 0
S = 2
I = 9
Iterations: 120960
Time taken: 2.7571747303009033
Press ENTER to exit. To try again, you need to close and reopen the program.
```

Gambar 3.3 Hasil keluaran dari program.

4) NUMBER + NUMBER = PUZZLE

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1.
Hasilnya adalah sebagai berikut.

```

Cryptarithmic Solver
===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test4
--- Solving Problem ---
Problem: NUMBER + NUMBER = PUZZLE

--- Results ---
1 solution(s) found.
N = 2
U = 0
M = 1
B = 6
E = 8
R = 9
P = 4
Z = 3
L = 7
Iterations: 2903040
Time taken: 1166.8649158477783
Press ENTER to exit. To try again, you need to close and reopen the program.

```

Gambar 3.4 Hasil keluaran dari program.

5) TOM + NAG = GOAT

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1. Hasilnya adalah sebagai berikut.

```

Cryptarithmic Solver
===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test14
--- Solving Problem ---
Problem: TOM + NAG = GOAT

--- Results ---
25 solution(s) found.
T = 3 T = 3 T = 3 T = 3 T = 4 T = 4 T = 4 T = 4 T = 4 T = 4 T = 6 T = 6 T = 6 T = 6 T = 6 T = 7 T = 7 T = 7 T = 7 T = 7 T = 8 T = 8 T = 8 T = 8 T = 8
O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0 O = 0
M = 2 M = 2 M = 2 M = 2 M = 2 M = 3 M = 3 M = 3 M = 3 M = 3 M = 5 M = 5 M = 5 M = 5 M = 5 M = 6 M = 6 M = 6 M = 6 M = 6 M = 7 M = 7 M = 7 M = 7 M = 7
N = 7 N = 7 N = 7 N = 7 N = 7 N = 7 N = 6 N = 6 N = 6 N = 6 N = 6 N = 4 N = 4 N = 4 N = 4 N = 4 N = 3 N = 3 N = 3 N = 3 N = 3 N = 2 N = 2 N = 2 N = 2 N = 2
A = 4 A = 5 A = 6 A = 6 A = 8 A = 9 A = 2 A = 5 A = 7 A = 8 A = 9 A = 2 A = 3 A = 7 A = 8 A = 9 A = 2 A = 4 A = 5 A = 8 A = 9 A = 3 A = 4 A = 5 A = 6 A = 9
G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1 G = 1
Iterations: 185840
Time taken: 6.311949968338013
Press ENTER to exit. To try again, you need to close and reopen the program.

```

Gambar 3.5 Hasil keluaran dari program.

6) AA + BB + CC = ABC

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1. Hasilnya adalah sebagai berikut.


```
Cryptarithmic Solver

===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test12
--- Solving Problem ---
Problem: AA + BB + CC = ABC

--- Results ---
1 solution(s) found.
A = 1
B = 9
C = 8
Iterations: 504
Time taken: 0.29592394828796387
Press ENTER to exit. To try again, you need to close and reopen the program.
```

Gambar 3.6 Hasil keluaran dari program.

7) NO + GUN + NO = HUNT

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1.
Hasilnya adalah sebagai berikut.'

```
Cryptarithmic Solver

===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test11
--- Solving Problem ---
Problem: NO + GUN + NO = HUNT

--- Results ---
1 solution(s) found.
N = 8
O = 7
G = 9
U = 0
H = 1
T = 2
Iterations: 105840
Time taken: 10.233978033065796
Press ENTER to exit. To try again, you need to close and reopen the program.
```

Gambar 3.7 Hasil keluaran dari program.

8) HERE + SHE = COMES

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1.
Hasilnya adalah sebagai berikut.

```
Cryptarithmic Solver
===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test8
--- Solving Problem ---
Problem:  HERE + SHE = COMES

--- Results ---
1 solution(s) found.
H = 9
E = 4
R = 5
S = 8
C = 1
O = 0
M = 3
Iterations: 423360
Time taken: 48.13191747665405
Press ENTER to exit. To try again, you need to close and reopen the program._
```

Gambar 3.8 Hasil keluaran dari program.

9) $EAT + THAT = APPLE$

Untuk memasukkannya, memiliki format yang sama seperti uji kasus 1.
Hasilnya adalah sebagai berikut.

```
Cryptarithmic Solver
===== CRYPTARITHMETIC SOLVER =====
This program is using a brute force algorithm, so it may take some time depending on your computer power.
You can add more than 1 (one) each file. Use `./folder_name/file_name` if it's on another directory.

Enter filename (.txt only): D:\IF Tubes Tucil\IF2211 Strategi Algoritma\Tucil 1 Cryptarithmic\test\test13
--- Solving Problem ---
Problem:  EAT + THAT = APPLE

--- Results ---
1 solution(s) found.
E = 8
A = 1
T = 9
H = 2
P = 0
L = 3
Iterations: 105840
Time taken: 7.1749794483184814
Press ENTER to exit. To try again, you need to close and reopen the program._
```

Gambar 3.9 Hasil keluaran dari program.