

Penyelesaian Cryptarithmic dengan Algoritma Brute Force

Diajukan untuk memenuhi tugas kecil matakuliah IF2211 Strategi Algoritma

Oleh:
13519036
Andrew



**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020**

A. Deskripsi Persoalan

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

Contoh persoalan:

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array} \qquad \begin{array}{r} 9 \ 5 \ 6 \ 7 \\ + \ 1 \ 0 \ 8 \ 5 \\ \hline 1 \ 0 \ 6 \ 5 \ 2 \end{array}$$

Sehingga S = 9, E = 5, N = 6, D = 7, M = 1, O = 0, R = 8, Y = 2.

B. Algoritma Brute-Force

Secara umum, penyelesaian Cryptarithmic dengan algoritma Brute-Force adalah sebagai berikut.

1. Program membaca file masukan, lalu mengidentifikasi operan dan huruf yang terdapat di dalam file.
2. Program melakukan permutasi angka 0-9, lalu menyimpan semua hasil permutasi ke dalam sebuah list.
3. Program kemudian mengisi value dari setiap huruf dengan masing-masing elemen di dalam list hasil permutasi angka 0-9.
4. Program mengecek apakah angka-angka yang dimasukkan merupakan solusi dari huruf-huruf yang berkorespondensi.
5. Jika program menemukan solusi angka-angka yang benar, program akan mengembalikan persoalan dengan masing-masing huruf sudah disubstitusi oleh angka-angka yang berkorespondensi.

C. Source Code

cryptarithm.py

```
#!/usr/bin/env python3
#-- Andrew 13519036 --#

# pengimpor sejati
import time

# permutasi
def perms(word):
    stack = list(word)
    results = [stack.pop()]
    while len(stack) != 0:
        c = stack.pop()
        new_results = []
        for w in results:
            for i in range(len(w)+1):
                new_results.append(w[:i] + c + w[i:])
        results = new_results
```

```

        return results

# mengecek kebenaran Cryptarithm
def isCrypto(angka, mapHuruf, kata, strip, waktu):
    i = 0
    ketemu = False
    while i < len(angka) and not ketemu:
        ruasKiri = 0
        ruasKanan = 0
        listKiri = []
        listKanan = []

        for j in range(len(mapHuruf)):
            mapHuruf[j][1] = int(angka[i][j])

        for j in range(len(kata)-1):
            for k in range(len(kata[j])):
                for l in range(len(mapHuruf)):
                    if mapHuruf[l][0] == kata[j][k]:
                        ruasKiri += mapHuruf[l][1]*(10**k)
                        break
                listKiri.append(ruasKiri)
                ruasKiri = 0

        for k in range(len(kata[len(kata)-1])):
            nolDepan = False
            for l in range(len(mapHuruf)):
                if mapHuruf[l][0] == kata[len(kata)-1][k]:
                    if k != len(kata[len(kata)-1])-1:
                        ruasKanan += mapHuruf[l][1]*(10**k)
                        break
                    else:
                        if mapHuruf[l][1] != 0:
                            ruasKanan += mapHuruf[l][1]*(10**k)
                            break
                        else:
                            nolDepan = True
                            break
            if nolDepan:
                ruasKanan = 0
                break
        listKanan.append(ruasKanan)

        if sum(listKiri) == sum(listKanan):
            print("\n")
            for ii in range(len(listKiri)):
                if ii != len(listKiri)-1:
                    print(str(listKiri[ii]))
            else:
                print(str(listKiri[ii])+"+")
            print(strip[0])
            print(listKanan[0])

            print("\nWaktu komputasi: "+str(time.time()-waktu)+" detik.")
            print("Total tes yang dilakukan: "+str(i+1)+".")
            ketemu = True
        else:
            i += 1
    return ketemu

# balikkan kata perhuruf
def reverseKata(kata):
    for e in kata:
        e.reverse()
    return

# fungsi utama

```

```

def main(filename):
    waktu = time.time()

    angka = perms("0123456789")

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

    lines = [line.strip() for line in f]

    for i in lines:
        print(i)

    huruf = []
    kata = []
    strip = []
    for e in lines:
        if e[0] != '-':
            if e[len(e)-1] != '+':
                kata.append(list(e))
            else:
                kata.append(list(e[:len(e)-1]))
        else:
            strip.append(e)
            listlines = list(e)
            for i in listlines:
                if (i not in huruf) and (i != '+') and (i != '-'):
                    huruf.append(i)

    if len(huruf) > 10:
        print("\nJumlah huruf lebih dari 10.")
        return

    mapHuruf = [[0 for j in range(2)] for i in range(len(huruf))]
    for i in range(len(huruf)):
        mapHuruf[i][0] = huruf[i]

    reverseKata(kata)

    if not isCrypto(angka, mapHuruf, kata, strip, waktu):
        print("\nSolusi tidak ditemukan.")
        print("Waktu komputasi: "+str(time.time()-waktu)+" detik.")
        return

# program utama
print("Input \"#\" untuk keluar.")
filename = input("Silakan masukkan nama file: ")
while filename != "#":
    main(filename)
    filename = input("silakan masukkan nama file: ")
print("Terima kasih telah menggunakan program ini.")

```

D. Screenshot Penyelesaian Soal

Input	Output
<pre>Silakan masukkan nama file: test1.txt SEND MORE+ ----- MONEY</pre>	<pre>9567 1085+ ----- 10652 Waktu komputasi: 60.7610285282135 detik. Total tes yang dilakukan: 2629936.</pre>
<pre>silakan masukkan nama file: test2.txt NUMBER NUMBER+ ----- PUZZLE</pre>	<pre>201689 201689+ ----- 403378 Waktu komputasi: 42.40549993515015 detik. Total tes yang dilakukan: 1348572.</pre>
<pre>silakan masukkan nama file: test3.txt CLOCK TICK TOCK+ ----- PLANET</pre>	<pre>90892 6592 6892+ ----- 104376 Waktu komputasi: 110.692387342453 detik. Total tes yang dilakukan: 3441112.</pre>
<pre>silakan masukkan nama file: test4.txt THREE THREE TWO TWO ONE+ ----- ELEVEN</pre>	<pre>84611 84611 803 803 391+ ----- 171219 Waktu komputasi: 58.10004758834839 detik. Total tes yang dilakukan: 1489535.</pre>
<pre>silakan masukkan nama file: test5.txt APOLITICAL PENICILLIN PICKPOCKET+ ----- KNICKKNACK</pre>	<pre>6182373462 1503432230 1349184957+ ----- 9034990649 Waktu komputasi: 48.337430477142334 detik. Total tes yang dilakukan: 712279.</pre>
<pre>silakan masukkan nama file: test6.txt IMPRESARIO PARATROOPS TETRAMETER+ ----- ASPIRATION</pre>	<pre>4237689745 3979175538 1617926167+ ----- 9834791450 Waktu komputasi: 17.634190320968628 detik. Total tes yang dilakukan: 273060.</pre>

<pre> silakan masukkan nama file: test7.txt COINCIDENCE ELECTRICIAN+ ----- ACCELERATOR </pre>	<pre> 18641635415 52510961674+ ----- 71152597089 </pre> <p>Waktu komputasi: 78.6536819934845 detik. Total tes yang dilakukan: 1433978.</p>
<pre> silakan masukkan nama file: test8.txt COMPROMISE STRETCHIEST MICROSCOPIC+ ----- HOMESTRETCH </pre>	<pre> 9317831045 42852960542 10983493709+ ----- 63154285296 </pre> <p>Waktu komputasi: 211.76504635810852 detik. Total tes yang dilakukan: 2979926.</p>

E. Alamat Repository

https://github.com/andrcyes/Tucil1_13519036

F. Checklist

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