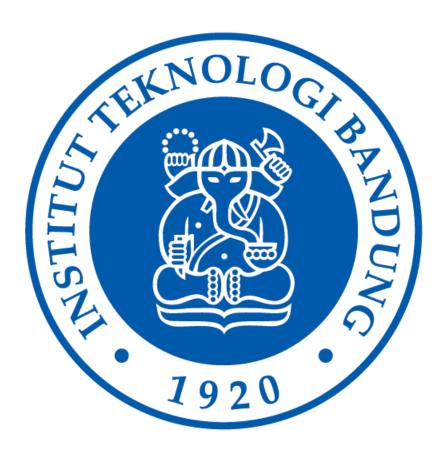
TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA PENYELESAIAN CRYPTARITHMETIC DENGAN ALGORITMA BRUTE FORCE



Disusun oleh:

Nama: Muhammad Furqon NIM: 13519184

PROGRAM STUDI TEKNIK INFORMATIKA SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG 2021

Laporan Tugas Kecil 1 Strategi Algoritma

Cryptarithmetic

- 1. Algoritma Brute Force
 - 1. Cari semua huruf yang unik atau berbeda yang terdapat pada semua operand
 - 2. Cari permutasi dari angka 0 sampai 9 dengan panjang sebanyak jumlah huruf yang unik
 - 3. Setiap huruf masing-masing diberi nilai, yaitu angka yang unik sesuai dengan salah satu permutasi
 - 4. Nilai tersebut diperiksa dari kolom atau digit yang paling kanan, konstrainnya adalah jumlah nilai operand yang di atas harus sesuai dengan yang di bawah, simpan *remainder* jika menghasilkan jumlah yang lebih besar dari sepuluh. Syarat lainnya digit paling kiri (huruf pertama) nilainya tidak boleh nol
 - 5. Ulangi langkah empat sampai semua persamaan diperiksa. Jika ada yang melanggar syarat ulangi langkah ketiga dengan permutasi yang berbeda

Contoh:

SEND

MORE+

MONEY

Diperiksa persamaan D+E = Y+remainder1

Diperiksa persamaan N+R+remainder1 = E+remainder2

Dan seterusnya sampai semua persamaan diperiksa, syarat lainnya S dan M (huruf terkiri) adalah nol. Jika terdapat syarat yang dilanggar ambil angka unik dari permutasi yang lain.

SourceCode(Python):

Tucil1.py

```
#Muhammad Furqon (13519184)
#Cryptarithmatic

import time
import os

last_hasil=0,0,0
length_max=0
remainder=0
angka=0
jmlh=0
valid=True
jmlh_loop=0
list_angka = [0,1,2,3,4,5,6,7,8,9]
```

```
list_huruf = []
correct =0
ada jawaban=False
#Permutasi elemen
def permutasi(elemen, r):
    n = len(elemen)
    indeks = list(range(n))
    perm = list(range(n, n-r, -1))
    yield tuple(elemen[i] for i in indeks[:r])
    stop=False
    while(not stop):
        stop=True
        for i in reversed(range(r)):
            perm[i] -= 1
            if perm[i] == 0:
                indeks[i:] = indeks[i+1:] + indeks[i:i+1]
                perm[i] = n - i
            else:
                j = perm[i]
                indeks[i], indeks[-j] = indeks[-j], indeks[i]
                yield tuple(elemen[i] for i in indeks[:r])
                stop=False
                break
#Referensi: https://stackoverflow.com/questions/104420/how-to-generate-all-
permutations-of-a-list
#Dictionary huruf unik
isMain=True
substring="+"
substring skip="-"
list main=[]
list_main_output=[]
list hasil=[]
dict_huruf = {}
#List untuk setiap kata, inisialisasi dictionary
os.chdir("...")
cur dir=os.getcwd()
os.chdir(cur_dir+"\\test")
nama_file = input("Masukkan nama file masukan: ")
cur dir=os.getcwd()
path=cur_dir+ "\\" + nama_file
file = open(path,"r")
Lines = file.readlines()
start = time.time()
```

```
for kata in Lines:
    skip=False
    kata=kata.strip()
    if(substring_skip in kata):
        skip=True
    if(not skip):
        if(isMain):
            if(substring in kata):
                isMain=False
                kata=kata.replace('+','')
            list_main.append(list(kata))
            list main output.append(list(kata))
        else:
            list hasil=list(kata)
        for huruf in kata:
            dict_huruf[huruf] = dict_huruf.get(huruf,0) + 1
#Buat dict kosong, list huruf
for key in dict huruf:
    list_huruf.append(key)
    dict_huruf[key] = -1
list_hasil_output=list(list_hasil)
#Cek dari kanan/last
for x in list main:
   x.reverse()
list hasil.reverse()
#Ujung kiri tidak boleh 0, angka mulai dari 1
#Length di sini adalah indeks ujung, len-1
length_main_max = len(max(list_main,key=len))-1
length max = len(list hasil)-1
# Permutasi
perm = permutasi(list_angka,len(dict_huruf))
for x in list(perm):
   #Setup
    correct=0
    angka=0
    valid=True
   #ubah nilai dictionary
   for y in x:
```

```
dict_huruf[list_huruf[angka]]=y
    angka += 1
while(i<=length_max and valid):</pre>
    if(i <= length main max):</pre>
        for sublist in list_main:
            if(i<=len(sublist)-1):</pre>
                 temp = dict huruf[sublist[i]]
                 if(i==(len(sublist)-1)):
                     if(temp==0):
                         valid=False
                         remainder=0
            else:
                 temp=0
            jmlh += temp
        jmlh += remainder
        if jmlh>=10:
            remainder=jmlh//10
        else:
            remainder=0
        if(i!=length_max):
            jmlh=jmlh%10
        #penjumlahan elemen
    else:
        jmlh=remainder
    #cek dengan elemen dari hasil
    last_hasil = dict_huruf[list_hasil[i]]
    if(i==(len(list_hasil)-1)):
        if(last hasil==0):
            valid=False
            remainder=0
    if(valid):
        if(jmlh==last_hasil):
            correct+=1
            if(correct==length_max+1):
                valid=False
                remainder=0
        else:
            valid=False
            remainder=0
    jmlh=0
    i+=1
if(correct==length_max+1):
    #Print soal
    count=0
```

```
print("Soal:")
        for output in list main output:
            selisih=length_max-len(output)+1
            print(" "*selisih,end="")
            for x in output:
                print(x, end="")
            if(count==len(list main output)-1):
                print("+",end="")
            print()
            count+=1
        print("-" * (length_max+1))
        for x in list hasil output:
            print(x, end="")
        print()
        #Print jawaban
        count=0
        print("Jawaban:")
        for output in list_main_output:
            selisih=length max-len(output)+1
            print(" "*selisih,end="")
            for x in output:
                print(dict_huruf[x], end="")
            if(count==len(list_main_output)-1):
                print("+",end="")
            print()
            count+=1
        print("-" * (length_max+1))
        for x in list_hasil_output:
            print(dict huruf[x], end="")
        print()
        #Print tes dan waktu
        print('Jumlah total tes: {}'.format(jmlh_loop))
        end = time.time()
        print('Waktu pengerjaan adalah {:.2f} s'.format(end-start))
        correct=0
        ada jawaban=True
    jmlh_loop += 1
if(not ada_jawaban):
    print("Tidak ditemukan solusi")
input("Tekan enter untuk mengakhiri program")
```

3. Skrinshut Program:

1.

2.

```
    test.txt

 Tucil1(Testing) > ≡ test1.txt
  1 NUMBER
   2 NUMBER+
   4 PUZZLE
Masukkan nama file masukan: test1.txt
Soal:
NUMBER
NUMBER+
PUZZLE
Jawaban:
201689
201689+
403378
Jumlah total tes: 728503
Waktu pengerjaan adalah 10.33 s
```

```
≡ test2.txt × ≡ test3.txt

    test2.txt

  1 TILES
  2 PUZZLES+
  4 PICTURE
Masukkan nama file masukan: test2.txt
Soal:
 TILES
PUZZLES+
PICTURE
Jawaban:
 91542
3077542+
3169084
Jumlah total tes: 3328706
Waktu pengerjaan adalah 19.46 s
```

```
Tucil1.py

≡ test3.txt
                                    X
   ≡ test3.txt
          CLOCK
          TICK
          TOCK+
           PLANET
     5
Masukkan nama file masukan: test3.txt
Soal:
CLOCK
 TICK
TOCK+
PLANET
Jawaban:
90892
 6592
 6892+
104376
Jumlah total tes: 3302474
Waktu pengerjaan adalah 20.37 s
```

```
Tucil1.py
             ≡ test4.txt ×

    test4.txt

   1 COCA
    2 COLA+
        OASIS
Masukkan nama file masukan: test4.txt
Soal:
COCA
 COLA+
OASIS
Jawaban:
 8186
8106+
16292
Jumlah total tes: 123694
Waktu pengerjaan adalah 0.67 s
```

6.

```
Tucil1.py

    test5.txt

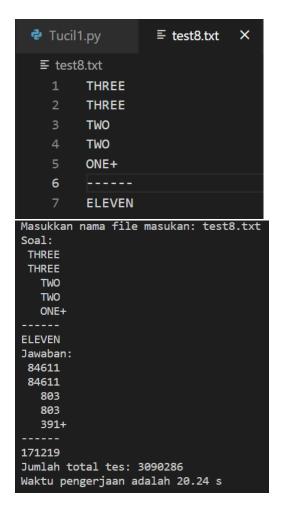
    test5.txt

          HERE
          SHE+
     4
          COMES
Masukkan nama file masukan: test5.txt
Soal:
HERE
 SHE+
COMES
Jawaban:
9454
 894+
10348
Jumlah total tes: 575301
Waktu pengerjaan adalah 2.63 s
```

```
Tucil1.py
                    ≡ test6.txt ×

    test6.txt
         DOUBLE
         DOUBLE
         TOIL+
         -----
    4
          TROUBLE
Masukkan nama file masukan: test6.txt
Soal:
DOUBLE
DOUBLE
 TOIL+
TROUBLE
Jawaban:
798064
798064
 1936+
1598064
Jumlah total tes: 2898675
Waktu pengerjaan adalah 15.45 s
```

```
Tucil1.py
                     ≡ test7.txt
                                  ×
   ≡ test7.txt
          NO
          GUN
          NO+
          HUNT
     5
Masukkan nama file masukan: test7.txt
Soal:
 NO
GUN
 NO+
----
HUNT
Jawaban:
 87
908
 87+
----
1082
Jumlah total tes: 134190
Waktu pengerjaan adalah 0.81 s
```



4. Link Drive:

https://drive.google.com/drive/folders/1AjK0GVVXSn9aJEk0EGZhuTJI9i0NTEWF?usp=sharing

Drive Pengumpulan

https://drive.google.com/drive/u/0/folders/1Ugq787WRmed5-ZR mZizJmjxCMRnfhlo

Poin	Ya	Tidak
1. Program berhasil		
dikompilasi tanpa	$\sqrt{}$	
kesalahan (no syntax error)		
2. Program berhasil <i>running</i>	$\sqrt{}$	
3. Program dapat membaca		
file		
masukan dan menuliskan	V	
luaran.		

4. Solusi cryptarithmetic		
hanya benar		
untuk persoalan		$\sqrt{}$
cryptarihtmetic		
dengan dua buah operand.		
5. Solusi <i>cryptarithmetic</i>		
benar untuk		
persoalan <i>cryptarihtmetic</i>	$\sqrt{}$	
untuk		
lebih dari dua buah operand.		

Referensi:

https://puzzling.stackexchange.com/questions/51/alphametic-verbal-arithmetic-general-strategy http://www.cryptarithms.com/default.asp?pg=1

https://people.revoledu.com/kardi/tutorial/CryptArithmetic/AttractionsIntentionsRegeneration.html https://stackoverflow.com/questions/104420/how-to-generate-all-permutations-of-a-list