**TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA**

**SEMESTER II TAHUN 2020/2021**

**Penyelesaian *Cryptarithmetic* dengan Algoritma *Brute Force***

Oleh

Arsa Daris Gintara – 13519037

A black and white drawing of a cartoon character

Description automatically generated with low confidence

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2020/2021**

# Algoritma *brute force*

1. Pertama program meminta masukkan string berupa file .txt yang terletak pada direktori /test/ untuk kemudian dibaca.

2. Kemudian akan dibagi 2 list, list pertama dicari karakter uniknya selain karakter ‘+’, ‘-‘, ‘ ‘, dan ‘\n’, list kedua untuk menampung semua karakter tiap barisnya.

3. Setelah itu akan dicari permutasi 0-9 sejumlah karakter unik disimpan pada sebuah list, contoh untuk kasus 2 karakter unik [[1,2],[2,1],[3,1],dst].

4. Lalu pada setiap kemungkinan permutasi akan dibuat pasangan antara karakter unik dengan permutasinya, contoh [[‘A’,1],[‘B’,2],dst]

5. Pasangan karakter unik ini kemudian digunakan untuk mengkonvert list awal yang masih berupa karakter(huruf) menjadi list integer.

6. Kemudian list integer dijumlah tiap operannya, jika penjumlahan tersebut tidak cocok dengan hasil penjumlahan (anggota terakhir dari list) maka akan dilanjutkan dengan kemungkinan permutasi lain (kembali ke langkah ke 4).

7. Jika penjumlahan tiap operan cocok dengan hasil penjumlahan maka akan ditampilkan hasilnya di layar, waktu eksekusi program, jumlah total tes, dan keluar dari iterasi pencocokan pasangan.

8. Jika setiap iterasi 4-6 tidak ditemui kecocokan, maka tidak ada kemungkinan permutasi yang cocok dan akan ditampilkan pesan ‘Tidak ada solusi yang memenuhi’.

# *Source Code* Program

import time

import time

*#list tiap line pada file*

def readFile(*filename*):

  stringList = []

  with open(filename,'r') as file:

    for line in file:

      stringList.append(line.strip('\n'))

  return stringList

*#mencari list character yang ada*

def charList(*filename*):

  stringnya = ''

  with open(filename,'r') as file:

    for line in file:

      stringnya += line.strip(' \n+-')

  return list(set(stringnya))

*#mencari kemungkinan-kemungkinan pasangan*

def genPermutasi(*theList*, *size*, *n*, *hasil*):

  if (size == 1):

    tampung = []

    for i in range(n):

      tampung = tampung + [theList[i]]

    hasil.append(tampung)

  for i in range(size):

    genPermutasi(theList, size-1, n, hasil)

    if (size%2 == 1):

      tmp = theList[0]

      theList[0] = theList[size-1]

      theList[size-1] = tmp

    else:

      tmp = theList[i]

      theList[i] = theList[size-1]

      theList[size-1] = tmp

*#checker pasangan sesuai atau tidak*

def checker(*theList*):

  jumlah = 0

  for i in range(len(theList)-2):

    jumlah += theList[i]

  if (jumlah == theList[len(theList)-1]):

    return True

  else:

    return False

*#main program*

file = input("Masukkan nama file yang terletak di folder /test/ (contoh: 1.txt) : ")

direktori = 'test/'+file

try:

  open(direktori)

except:

  direktori = '../test/'+file

try:

  open(direktori)

except:

  direktori = 'Tucil1\_13519037/test/'+file

initList = readFile(direktori)

Chara = charList(direktori)

initTime = time.perf\_counter()

permutasi = []

genPermutasi([0,1,2,3,4,5,6,7,8,9], 10, len(Chara), permutasi)

counter = 0

for perm in permutasi:

  counter += 1

  convList = []

*#pasangan angka dan huruf*

  pair = []

  for i in range(len(Chara)):

    pair.append([Chara[i],perm[i]])

  for line in initList:

    angka = 0

    lineCleaned = line.strip(' \n+-')

    for c in lineCleaned:

      isFZero = False

      for cari in range(len(pair)):

        if (c == pair[cari][0]):

          if (c == lineCleaned[0] and pair[cari][1] == 0):

            isFZero = True

          break

      if(isFZero):

        break

      angka = angka\*10 + pair[cari][1]

    if(isFZero):

      break

    if ('-' in line):

      convList.append(line)

    else:

      convList.append(angka)

*#pasangan sesuai ketemu*

  if (not(isFZero) and checker(convList)):

    for line in initList:

      print(line.rjust(9))

    print()

    for lineConv in range(len(convList)):

      if (lineConv == len(convList)-3):

        print(('+ ' + str(convList[lineConv])).rjust(9))

      else:

        print(str(convList[lineConv]).rjust(9))

    print('Waktu eksekusi program', time.perf\_counter()-initTime, 'detik')

    print('Jumlah total tes', counter, 'kali')

    break

else:

  print('Tidak ada solusi yang memenuhi')

# *Screenshot* Program

|  |  |
| --- | --- |
| Input | Output |
| Graphical user interface, text, application, Word  Description automatically generated | Text  Description automatically generated |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |
|  | Text  Description automatically generated |

|  |  |
| --- | --- |
| Graphical user interface, text, application, table  Description automatically generated with medium confidence |  |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |
| Graphical user interface, text  Description automatically generated |  |
| Graphical user interface, text, application  Description automatically generated | Text  Description automatically generated |

# Alamat Program

<https://github.com/arsa-dg/cryptarithmetic-bruteforce>

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi tanpa kesalahan (no syntax error) | √ |  |
| 1. Program berhasil *running* | √ |  |
| 1. Program dapat membaca file masukan dan menuliskan luaran. | √ |  |
| 1. Solusi *cryptarithmetic* hanya benar untuk persoalan *cryptarithmetic* dengan dua buah *operand.* |  | √ |
| 1. Solusi *cryptarithmetic* benar untuk persoalan *cryptarithmetic* untuk lebih dari dua buah operand | √ |  |