

Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2020/2021

Penyelesaian *Cryptarithmic* dengan Algoritma *Brute Force*



Disusun Oleh:

Ryo Richardo / 13519193

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2021

1. Algoritma *Brute Force*

Deskripsi Langkah-langkah penyelesaian permasalahan *Cryptarithmic* dengan algoritma *Brute Force* adalah sebagai berikut:

- i. Menginisialisasi *array* untuk menyimpan jumlah setiap huruf sesuai dengan posisi digitnya (misalnya huruf N pada posisi ratusan akan bernilai 100). *Array* ini akan disebut sebagai *array* huruf.
- ii. Menginisialisasi *array* untuk menyimpan maksimal 10 huruf yang nilainya bersesuaian dengan indeks *array* huruf. *Array* ini akan disebut sebagai *array* kunci.
- iii. Melakukan input permasalahan *Cryptarithmic* yang ingin diselesaikan.
- iv. Menyimpan setiap operan dalam *array* baru dan hasil dalam sebuah variabel.
- v. Menyimpan jumlah setiap huruf dari operan dan hasil yang bersesuaian dengan posisi digitnya (setiap huruf pada variabel hasil akan bernilai negatif) pada *array* huruf.
- vi. Menyimpan semua indeks *array* huruf yang elemennya tidak nol pada *array* kunci.
- vii. Melakukan permutasi pada *array* kunci.
- viii. Setiap hasil permutasi baru pada *array* kunci akan diperiksa dengan cara menjumlahkan hasil perkalian indeks *array* kunci dengan elemen *array* huruf yang bersesuaian.
- ix. Jika hasil penjumlahannya adalah nol, maka akan diperiksa untuk setiap huruf pertama operan dan hasil agar tidak bernilai nol.
- x. Jika setiap huruf pertama operan dan hasil tidak bernilai nol, maka memberikan output angka-angka yang bersesuaian dengan format yang sama seperti input.

2. *Source program*

```
# Program Cryptarithmic
# Ryo Richardo 13519193

# Global
count = 0
start = 0

# fungsi rekursif untuk mencari permutasi
def permut(a, x):
```

```

global count, start
if (x == 1):

    # memeriksa jumlah dari perkalian setiap huruf dengan kodenya
    sum = 0
    for i in range (10):
        sum += i * huruf[kunci[i]]

    # jumlah percobaan
    count += 1

    # jika jumlah operan = jumlah hasil, maka benar
    if (sum == 0):
        bol = True

    # memeriksa digit pertama apakah 0
    for kata in word:
        if (kunci[0] == ord(kata[0])-64):
            bol = False
    if (kunci[0] == ord(hasil[0])-64):
        bol = False

    # jika digit pertama tidak 0, print dan exit program
    if bol:
        # menyimpan angka kode tiap huruf
        for i in range (10):
            huruf[kunci[i]] = i

        # print semua operan kecuali operan terakhir
        for j in range (len(word)-1):
            for i in range (len(word[j])):
                print(huruf[ord(word[j][i])-64], end="")
            print()

        # print operan terakhir
        for char in word[len(word)-1]:
            print(huruf[ord(char)-64], end="")
        print("+")

        # print garis dan hasil penjumlahan
        print(garis)
        for char in hasil:
            print(huruf[ord(char)-64], end="")

        # print jumlah percobaan dan waktu program berjalan
        print("\nJumlah percobaan =", count, "kali")
        print("Waktu eksekusi   =", time.time() - start, "detik")

```

```

        exit(0)

# mencari pola permutasi yang baru
for i in range(x):
    permut(a, x-1)
    if (x % 2 == 1):
        a[0], a[x-1] = a[x-1], a[0]
    else:
        a[i], a[x-1] = a[x-1], a[i]

# MAIN PROGRAM
import time

# inisialisasi array huruf dan kunci
huruf = [0 for i in range (27)]
kunci = [0 for i in range (10)]

# input nama dan membuka file
nama = str(input("Masukkan nama file yang berada dalam folder test: "))
file = open('../test/' + nama + '.txt', 'r')
lines = file.readlines()

# inisialisasi array untuk menyimpan semua operan
word = ["" for i in range (len(lines) - 2)]

# mengisi array dengan operan (kecuali operan terakhir)
for i in range (len(lines) - 3):
    word[i] = lines[i].strip()

# mengisi array dengan operan terakhir
word[len(lines) - 3] = lines[len(lines) - 3].strip("+\n")

# variabel garis dan hasil
garis = lines[len(lines) - 2].strip()
hasil = lines[len(lines) - 1].strip()

# memulai countdown
start = time.time()

# mengisi kamus huruf yang terpakai ke array huruf.
for kata in word:
    for i in range (len(kata)):
        huruf[ord(kata[i])-64] += 10**(len(kata)-i-1)

for i in range (len(hasil)):
    huruf[ord(hasil[i])-64] -= 10**(len(hasil)-i-1)

# mengisi indeks huruf ke array kunci untuk dipermutasi

```

```

idx = 0
for i in range (0, 26):
    if (huruf[i] != 0):
        kunci[idx] = i
        idx += 1

# melakukan permutasi hingga exit program
permut(kunci, 10)

```

3. Contoh *input* dan *output*

No	Input	Output
1	<pre> 1 SEND 2 MORE+ 3 ----- 4 MONEY </pre>	<pre> 9567 1085+ ----- 10652 Jumlah percobaan = 2641562 kali Waktu eksekusi = 9.501362800598145 detik </pre>
2	<pre> 1 HERE 2 SHE+ 3 ----- 4 COMES </pre>	<pre> 9454 894+ ----- 10348 Jumlah percobaan = 1177856 kali Waktu eksekusi = 4.244372844696045 detik </pre>
3	<pre> 1 CLOCK 2 TICK 3 TOCK+ 4 ----- 5 PLANET </pre>	<pre> 90892 6592 6892+ ----- 104376 Jumlah percobaan = 789453 kali Waktu eksekusi = 3.7008163928985596 detik </pre>
4	<pre> 1 THREE 2 THREE 3 TWO 4 TWO 5 ONE+ 6 ----- 7 ELEVEN </pre>	<pre> 84611 84611 803 803 391+ ----- 171219 Jumlah percobaan = 1567089 kali Waktu eksekusi = 5.831986427307129 detik </pre>

5	<pre> 1 COCA 2 COLA+ 3 ----- 4 OASIS </pre>	<pre> 8186 8106+ ----- 16292 Jumlah percobaan = 1355041 kali Waktu eksekusi = 5.488498687744141 detik </pre>
6	<pre> 1 DOUBLE 2 DOUBLE 3 TOIL+ 4 ----- 5 TROUBLE </pre>	<pre> 798064 798064 1936+ ----- 1598064 Jumlah percobaan = 2193247 kali Waktu eksekusi = 8.563073635101318 detik </pre>
7	<pre> 1 NO 2 GUN 3 NO+ 4 ----- 5 HUNT </pre>	<pre> 87 908 87+ ----- 1082 Jumlah percobaan = 609843 kali Waktu eksekusi = 2.3052444458007812 detik </pre>
8	<pre> 1 MEMO 2 FROM+ 3 ----- 4 HOMER </pre>	<pre> 8485 7358+ ----- 15843 Jumlah percobaan = 238045 kali Waktu eksekusi = 0.8864607810974121 detik </pre>

4. Alamat *drive*

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

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.	✓	