Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II Tahun 2020/2021

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

Nama : Daru Bagus Dananjaya

NIM : 13519080

Kelas : K02

**Pendahuluan**

*Cryptarithmetic* (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 dan solusinya :



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

**Deskripsi Langkah-Langkah Algoritma Brute Force**

Pada persoalan ini, saya mendapatkan 2 buah metode *brute force* untuk mendapatkan solusi dari *Cryptarithmetic.* Pada kedua metode tersebut terdapat *data structure* berupa array/list dan juga dictionary. Array berfungsi untuk menyimpan *operand* dan hasil dari persoalan yang diberikan dan juga digunakan untuk menyimpan *operand* dan hasil yang sudah diubah menjadi representasi angka yang bersesuaian untuk dicek kesesuaiannya, sedangkan dictionary digunakan untuk menyimpan kamus berupa pasangan huruf dan angka yang merepresentasikannya.

Metode ini didasarkan pada konsep permutasi, terdapat maksimum 3628800 buah cara untuk menyusun 10 buah angka untuk merepresentasikan 10 buah huruf. Jika pada persoalan jumlah huruf kurang dari 10, maka metode ini tetap akan menghasilkan 3628800 buah cara untuk menyusun representasi angka, namun diambil sebagian saja sesuai jumlah huruf, hal ini tetap akan menghasilkan sebuah solusi walaupun tidak begitu efisien karena pasti ada yang berulang.

Berikut adalah langkah-langkahnya :

1. Baca file yang berisi persoalan *Cryptarithmetic* yang ingin diselesaikan, lalu *parsing* setiap *operand* dan juga *hasil*, ke dalam suatu list untuk diproses kemudian.
2. Pada bahasa pemograman Python, terdapat metode penyimpanan data yaitu Dictionary -pemilihan Dictionary daripada List didasarkan kepada efisiensi waktu pencarian elemen- yang akan digunakan untuk menyimpan setiap huruf yang ada pada *operand* dan juga hasil. Kemudian inisialisasi angka yang merepresentasikan huruf dengan angka sembarang, dalam hal ini saya memilih -1.
3. *Generate* semua kemungkinan permutasi 10 angka disusun 10, lalu simpan semua kemungkinan tersebut ke dalam suatu array/list. Cara meng-*generate* permutasi tersebut menggunakan iterasi *for loop* yang di-*nested* sebanyak 10 kali untuk setiap kemungkinan angka yang mungkin menduduki suatu urutan.
4. Mulai proses untuk pencocokan antara setiap kemungkinan permutasi 10*p*10 (diambil sebagian saja sesuai dengan jumlah huruf) dengan huruf yang direpresentasikan dengan angka pada permutasi tersebut. Cara pencocokannya adalah dengan mengubah setiap huruf menjadi angka yang merepresentasikannya, lalu hitung apakah jika setiap *operand* dijumlahkan akan menghasilkan hasil yang sesuai, jika sesuai maka berhenti proses pencocokan dan *Cryptarithmetic* berhasil terpecahkan, jika tidak maka ulang terus proses hingga hasil yang sesuai didapatkan atau semua permutasi telah diproses cocokan, jika tidak berhasil ditemukan maka *Cryptarithmetic* tidak berhasil diselesaikan.