

Hashing - Division Hash String Key (A12)

Dilakukan hashing terhadap sebuah String yang terdiri dari n karakter ke dalam tabel dengan M slot. String tersebut diperlakukan sebagai bilangan berbasis 256, dan hashing dilakukan dengan division method. Berikut ini adalah langkah-langkah untuk merepresntasikan String menjadi bilangan berbasis 256:

1. Untuk setiap karakter pada String S , ubahlah tiap karakter ke dalam ASCII.
Misalnya String yang akan diubah adalah DESAIN, maka setiap karakter tersebut diubah ke dalam ASCII.
2. Setelah mengetahui kode ASCII untuk semua karakter di dalam String S , langkah selanjutnya adalah mengkonversinya ke dalam bilangan berbasis 256 dengan rumus:
$$S_{256} = S[0] * 256^{n-1} + S[1] * 256^{n-2} + S[2]256^{n-3} + \dots + S[n-1]256^0$$

dengan:
 - o $S[i]$ adalah kode ASCII untuk karakter ke- i di dalam String.
 - o n adalah panjang String.
 - o i adalah indeks dari String, dimulai dari 0.

Contoh:

String: DESAIN

Kode ASCII D=68, E=69, S=83, A=65, I=73 dan N=78. Maka String desain dalam bentuk bilangan berbasis 256 adalah:

$$S = 68 * 256^5 + 69 * 256^4 + 83 * 256^3 + 65 * 256^2 + 73 * 256^1 + 78 * 256^0$$

$$S = 75.064.540.219.726$$

Dari contoh terlihat bahwa nilai S sangat besar untuk String yang panjangnya hanya 6 karakter. Hal ini dapat menyebabkan overflow jika kita menggunakan tipe data primitif seperti int atau long int. Menggunakan tipe data big integer bukanlah solusi yang baik, karena perhitungan akan jauh lebih lambat. Ada cara yang lebih cepat dan efisien, yaitu dengan memanfaatkan sifat-sifat operasi modulus. Perhatikan rumus berikut ini:

$$h(S) = (S[0] * 256^{n-1} + S[1] * 256^{n-2} + S[2]256^{n-3} + \dots + S[n-1]256^0) \bmod M$$

Kita tahu bahwa operasi modulus bersifat distributif:

$$\begin{aligned}(a + b) \bmod M &= (a \bmod M + b \bmod M) \bmod M \\ (a * b) \bmod M &= (a \bmod M * b \bmod M) \bmod M\end{aligned}$$

Sehingga fungsi $h(S)$ dapat dijabarkan menjadi:

$$\begin{aligned}h(S) &= ((S[0] * 256^{n-1}) \bmod M + (S[1] * 256^{n-2}) \bmod M + (S[2]256^{n-3}) \bmod M + \dots \\ &\quad + (S[n-1]256^0) \bmod M) \bmod M\end{aligned}$$

Setiap $S[i]$ dan 256^i juga dapat di mod dahulu dengan M sebelum dioperasikan dengan perkalian atau penjumlahan berikutnya. Bahkan untuk menghitung 256^i juga perlu di-mod, supaya tidak overflow.

$$\begin{aligned}h(S) &= ((S[0] \bmod M * 256^{n-1} \bmod M) \bmod M + (S[1] \bmod M * 256^{n-2} \bmod M) \bmod M) \bmod M \\ &\quad + \dots\end{aligned}$$

Implementasikanlah fungsi hash untuk key berupa String dengan cara modular seperti yang dijelaskan di atas dengan $M = 31991$! Anda tidak diwajibkan membuat kelas hash secara lengkap.

Spesifikasi Input:

Input dimulai dengan sebuah bilangan n , $1 \leq n \leq 100$ yang menyatakan banyaknya kasus yang akan diperiksa. Sebanyak n baris berikutnya, input dilanjutkan dengan key berupa String yang akan di-hash. String hanya mengandung huruf kecil dan kapital saja (tidak ada spasi, tanda baca, dll), dan panjangnya tidak lebih dari 1000 karakter.

Spesifikasi Output:

Untuk masing-masing kasus, keluarkan nilai hash dari String yang diinputkan.

Contoh Input:

```
8
DESAIN
ANALISIS
DAA
ALGORITMA
desain
analisis
daa
algoritma
```

Contoh Output:

```
13052
3381
26404
9200
5428
579
20374
27713
```