

Laporan Penjelasan Ide Tugas Pemrograman 1 SDA

Pengambilan Input

Dalam pengambilan input TP 1 kali ini, saya menggunakan metode yang sama seperti yang telah diberikan pada saat pengerjaan WS2-Selasa.

Struktur Data yang Digunakan

Dalam TP1 ini saya merepresentasikan toko dengan class Toko yang memiliki nama toko dan HashMap berisi donat didalamnya. Saya juga merepresentasikan donat dengan class Donat yang memiliki atribut jenis donat, jumlah stok donat dalam toko tersebut, serta jumlah chip yang ada di donat tersebut. Saya juga membuat HashMap di world untuk menyimpan semua toko yang ada. Saya juga membuat ArrayList integer berisi cara untuk mencapai target setiap harinya dan ArrayList tambahan untuk menyimpan semua stok dan chip donat yang tersedia di hari tertentu.

Pengambilan Informasi Toko dan Jenis Donat yang ada

Untuk pengambilan informasi Toko dan Donat, saya loop sebanyak jumlah toko yang tersedia lalu saya buat objek toko dengan nama toko tersebut dan saya masukkan semua donat sesuai informasi yang didapat dari input ke dalam HashMap donat yang dimiliki toko tersebut.

Kompleksitas : $O(\text{jumlah toko} * \text{jumlah jenis donat per toko})$

Proses Penghitungan Kemungkinan

Informasi selanjutnya yang didapatkan adalah jumlah hari, lalu saya melakukan For Loop sebanyak banyak hari yang diminta, lalu melakukan semua proses yang terjadi setiap hari.

- Mengambil input tentang berapa toko yang buka pada hari itu dan membuat 2 ArrayList Integer yang nantinya akan diisi jumlah chip tiap donat dan jumlah stok tiap donat. Lalu melakukan For Loop sebanyak N toko yang buka dan menjalankan method open() untuk mengganti value atribut open dari toko tersebut dari false menjadi true. Selain itu, memasukkan jumlah chip dan jumlah stok dari tiap donat yang ada di toko yang buka tersebut agar termasuk saat penghitungan kemungkinan mendapatkan chocochip untuk target setiap harinya.

Kompleksitas : $O(\text{jumlah toko})$

- Method pertama dan utama adalah method Target. Method ini menerima 5 parameter, yaitu ArrayList berisi jumlah chip setiap donat, ArrayList berisi stok donat, target, 1 Integer yang bertugas sebagai pointer dan 1 array multidimensi untuk memoization. Method ini merupakan method rekursif, base casenya adalah :
 1. Jika target = 0, maka return 1, tidak melakukan apa-apa.
 2. Jika target < 0, maka tidak ada yang bisa dilakukan (return 0).
 3. Jika target > 0 tetapi panjang chip donat dan stok donat = 0, maka tidak ada yang bisa dilakukan (return 0)

4. Jika hasil penghitungan jumlah kemungkinan untuk mencapai target dengan donat tersebut sudah pernah dihitung, return jumlah tersebut
5. Jika pointer sudah mencapai index terakhir dari jumlah donat, maka rekursi selesai (return 0)

Rekursif case yang digunakan adalah selain itu, menjalankan rekursif yang serupa dengan mencari kombinasi. ArrayList donat diiterasi dari pointer ke index 0 hingga abis untuk mendapatkan chip sesuai target, lalu di mod $10^9 + 7$ agar tidak overflow.

Kompleksitas : $O(\text{target} * \text{jumlah donat yang tersedia} * \text{stok setiap donat})$

- Setelah method target, selanjutnya method Duar dengan 3 parameter, yaitu nama toko, jenis donat, dan jumlah donat yang meledak. Setelah me-loop sejumlah banyaknya toko, saya menjalankan Duar dengan mengambil 3 input selanjutnya. Dalam method Duar, saya menjalankan method boom pada donat tersebut yang berfungsi mengurangi stok donat tersebut pada toko. Setelah dikurangi, donat tersebut diperiksa jika stoknya sudah 0, maka donat tersebut dihapus dari HashMap agar toko bisa menerima transfer dengan jumlah chip berbeda.

Kompleksitas : $O(\text{jumlah meledak} * (\text{jumlah toko} + \text{jumlah donat}))$

- Setelah selesai dengan duar, selanjutnya adalah menjalankan method Restock. Setelah me-loop sejumlah berapa restock yang dilakukan, saya mengambil 4 input yang digunakan dalam method isiUlang, yaitu nama toko, jenis donat, jumlah yang ingin di restock dan jumlah chip. Dalam method isiUlang, saya memeriksa adakah donat tersebut di toko yang tertera, jika tidak ada maka saya memasukkan donat baru ke dalam HashMap toko tersebut sesuai dengan informasi yang diberikan. Jika sudah ada, maka dilakukan pemeriksaan terhadap jumlah chip, jika chipnya sama maka akan dijalankan method tambah pada donat tersebut untuk menambah jumlah stok donat tersebut, jika jumlah chip berbeda maka akan gagal.

Kompleksitas : $O(\text{jumlah restock} * (\text{jumlah toko} + \text{jumlah donat}))$, karena prinsipnya sama dengan Duar

- Method terakhir adalah method Transfer yang menerima 4 input, toko asal donat, toko akhir donat, jenis donat, dan jumlah donat yang di transfer. Dalam method ini pertama saya memeriksa apakah di toko tujuan ada donat dengan jenis tersebut, jika tidak saya mengurangi donat di toko asal dengan method Duar, lalu menambahkan donat baru ke HashMap toko tujuan. Jika di toko tujuan sudah ada donat sejenis, maka saya memeriksa apakah kedua donat di toko berbeda itu memiliki chip yang sama. Jika sama, maka saya mengurangi jumlah donat tersebut di toko asal dengan method Duar kembali dan menambah donut di toko tujuan dengan method tambah. Jika kedua donat tersebut memiliki jumlah chip yang berbeda, maka transfer akan gagal.

Kompleksitas : $O(\text{jumlah transfer} * (\text{jumlah toko} * 2 + \text{jumlah donat toko awal} + \text{jumlah donat toko akhir}))$

- Setelah semua method selesai dikerjakan ,saya menutup semua toko dan men-clear semua ArrayList berisi jumlah stok dan chocochip donat yang tersedia.

Kompleksitas : $O(\text{jumlah toko})$