

Adhoc + Prefix Sum (Kalau Sempet)

By : Abdan Hafidz

Adhoc ~~Soal Ez~~

Ad hoc problems are those whose algorithms do not fall into standard categories with well-studied solutions. Each ad hoc problem is different; **no specific or general techniques exist to solve them.**

(USACO 1.2)

Biasanya soal – soal adhoc bercirikan :

- Suka dikerjakan di awal (gatau sih katanya begito)
- Keliatannya sederhana
- Tidak terlalu berpaku pada teori
- Sangat intuitif
- PENUH TRIK TIPU DAYA, MUSLIHAT, DAN JEBAKAN BATMAN
- NGESELIN, MENYIKSA DAN MEMAKSA kita supaya optimize kompleksitas kalau bisa jadi constant.

Adhoc ~~Soal Ez~~

Kategori pasangan soal adhoc biasanya terurut berdasarkan paling sering (constructive, math) sisanya sedikit seperti (geometry, bitwise, graph).

Soal adhoc math biasanya nguji kita tentang beberapa teori bilangan yang tricky

Soal adhoc konstruktif, bukan hanya mencari jawaban dari suatu persoalan, tetapi juga mencari konfigurasi yang menghasilkan jawaban tersebut. Konfigurasi yang dimaksud bisa berupa array, grid, kumpulan parameter, langkah-Langkah mencapai jawaban, dan lain-lain

Funfact soal – soal aneh seperti interactive, output only, dan batch juga sering pakai tag adhoc terutama constructive.

<https://tlx.toki.id/problems/osn-2019/2B>

Adhoc Soal Ez

Menghitung Angka

Batas Waktu : 1 detik

Batasan memori : 256 mb

Deskripsi Soal

Pak Denglek diberikan beberapa angka a dan b . Ia ingin mencari bilangan terkecil yang memenuhi i dan bilangan terbesar yang memenuhi j sehingga :

- Jika a dan b dibagi j maka bersisa sama dengan 0
- Jika a dikalikan dengan suatu bilangan asli k dan b dikalikan dengan suatu bilangan asli l maka keduanya memiliki hasil yang sama.

Dalam kasus ini anda diminta untuk menghitung berapakah hasil dari

$$ij - \frac{ij}{ab}$$

Format Masukan

Satu baris berisikan bilangan bulat a dan b .

Format Keluaran

Jawaban dari soal sesuai dengan yang diminta.

Batasan

$$1 \leq a, b \leq 10^9$$

Constructive

Algoritma konstruktif adalah algoritma yang bukan hanya mencari jawaban dari suatu persoalan, tetapi juga mencari konfigurasi yang menghasilkan jawaban tersebut. Konfigurasi yang dimaksud bisa berupa array, grid, kumpulan parameter, langkah-langkah mencapai jawaban, dan lain-lain

Contoh Soal

Deskripsi Soal: Diberikan sebuah bilangan bulat positif n . Carilah empat bilangan bulat positif a , b , c , dan d manapun sehingga $a + b + c + d = n$ dan $\text{FPB } a, b = \text{KPK } c, d$.

- Perhatikan bahwa FPB dari 1 dan bilangan apapun sama dengan 1
- Sementara itu, pasangan bilangan yang KPKnya 1 hanya 1 dan 1
- Jadi, kita bisa memilih $a = 1$, $b = n - 3$, $c = 1$, dan $d = 1$

- Soal ini membutuhkan algoritma konstruktif, karena kita diminta mengonstruksi kumpulan bilangan yang memenuhi syarat pada soal
- Biasanya, untuk menyelesaikan soal yang membutuhkan algoritma konstruktif, diperlukan suatu properti khusus
- Apa properti dari FPB dan KPK?

Prefix Sum

Misalkan kita mempunyai sebuah array sebanyak n elemen
Lalu kita diberi Q operasi, di mana setiap operasi ke i kita diminta untuk menjumlahkan berapa hasil jumlah elemen array dari range index ke 0 s/d R_i ?

Secara naif mungkin kita akan menghitung pada masing – masing operasi dengan menggunakan bruteforce, worst case setiap operasi menghitung sebanyak $R = n$, karena ada Q operasi maka kompleksitasnya $O(Qn)$.

Kita bisa optimasi dengan menggunakan metode prefix sum sehingga worst complexitynya adalah $O(n)$.

Note that :

```
pref[i] = arr[0] + arr[1] + ... + arr[i - 1] + arr[i]  
pref[i - 1] = arr[0] + arr[1] + ... + arr[i - 1]  
pref[i] = pref[i - 1] + arr[i]
```

Prefix Sum

Bagaimana jika diminta hasil sum dari rentang indeks Li sampai Ri?

$$\text{sum}(L, R) = \text{sum}(1, R) - \text{sum}(1, L - 1)$$

$$\text{sum}(L, R) = \text{pref}[R] - \text{pref}[L - 1]$$