Soal Competitive Programming (Final)

Soal 1: Jalur Terpendek

Sebuah kota memiliki N persimpangan dan M jalan satu arah. Amir ingin pergi dari persimpangan `1` ke persimpangan `N` dengan rute terpendek. Namun, Amir tidak suka melalui dua jalan berturut-turut dengan label jenis jalan yang sama (misalnya, dua jalan bertipe "A" berturut-turut). Setiap jalan bertipe "A" atau "B".

Input:

- Baris pertama berisi dua bilangan bulat N dan M.
- `M` baris berikutnya berisi: `u v t w` artinya ada jalan dari `u` ke `v` bertipe `t` (`A` atau `B`) dengan waktu tempuh `w`.

Output:

- Cetak waktu minimum dari simpul 1 ke simpul N dengan syarat Amir tidak boleh melewati dua jalan berturut-turut dengan tipe sama. Jika tidak bisa, cetak `-1`.

Contoh Input:

```
4 5
1 2 A 5
2 3 B 6
3 4 A 2
1 3 B 10
2 4 A 8
```

Contoh Output:

```
13
```

Penjelasan:

Rute terbaik adalah 1 → 2 → 3 → 4 dengan tipe jalan: A → B → A.

Soal 2: Penebangan Pohon Minimal

Dalam peta hutan N x M, sedangkan T adalah pohon, dan '**.**' adalah tanah kosong. Pemerintah ingin menebang pohon agar terbentuk jalur lurus horizontal atau vertikal dari (0, 0) ke (N-1, M-1). Tebangan pohon minimal.

Input:

- Baris pertama: N M
- Diikuti N baris string panjang M (T atau **.**)

Output:

- Cetak jumlah minimal pohon yang harus ditebang agar bisa membuat satu jalur lurus dari kiri atas ke kanan bawah.

Contoh Input:

```
3 3
T . T
T T T
. . T
```

Contoh Output:

```
2
```

Penjelasan:

Dari (0,0) ke (2,2) membutuhkan penebangan yang paling minimal adalah 2 T.

Soal 3: Optimasi Palindromic Subsequence

Diberikan sebuah string S dengan N karakter. Anda dapat melakukan operasi delete karakter dengan cost tertentu untuk setiap posisi. Tujuan adalah membuat palindromic subsequence terpanjang dengan total cost yang minimum.

Penjelasan

- String S dengan N karakter
- Setiap posisi i memiliki cost[i] untuk menghapus karakter di posisi tersebut
- Cari subsequence palindrom terpanjang dengan meminimalkan total cost penghapusan
- Jika ada beberapa solusi dengan panjang sama, pilih yang cost-nya minimum

Input
Baris 1: N (panjang string)
Baris 2: S (string input)
Baris 3: cost1 cost2 ... costN (cost penghapusan setiap posisi)

Output
Panjang maksimal palindrom dan total cost minimum

Contoh Input

```
8
abcdcbae
1 3 2 4 5 2 1 3
```

Contoh Output

```
5 6
```

Penjelasan

1. String: "abcdcbae", cost: [1,3,2,4,5,2,1,3]
2. Palindromic subsequence terpanjang: "abcba" atau "acdca"
3. Untuk "abcba": gunakan posisi 1,2,4,6,8 → tidak ada penghapusan di posisi ini
4. Hapus posisi 3,5,7 → cost = 2+5+1 = 8
5. Untuk "acdca": gunakan posisi 1,3,4,5,8 → hapus posisi 2,6,7 → cost = 3+2+1 = 6
6. Output: panjang 5, cost minimum 6

Soal 4: Kompresi String Berulang

Diberikan string S yang terdiri dari huruf kecil. Anda perlu mengompresi string dengan mengganti setiap kemunculan berulang dari substring dengan panjang minimum yang menghasilkan string terkompresi terpendek. Misalnya, "aababa" bisa dikompresi menjadi "aa2(ba)" jika "ba" diulang.

Input:
Baris pertama: string S ($1 \leq |S| \leq 10^5$), hanya huruf kecil.

Output:
Panjang string terkompresi terpendek.

Contoh Input:

```
aababa
```

Contoh Output:

```
5
```

Penjelasan Proses:
String "aababa" dapat dikompresi:

- Substring "ba" muncul dua kali (posisi 3-4 dan 5-6).
- Kompresi menjadi "aa2(ba)".

Panjang string terkompresi: |"aa2(ba)"| = 2 + 1 + 2 = 5 .
Output adalah panjang string terkompresi terpendek, yaitu 5.

Soal 5: Dekripsi Pesan Modulo

Anda diberikan pesan terenkripsi berupa array bilangan bulat A sepanjang N, yang dienkripsi menggunakan kunci K dengan operasi modulo. Setiap elemen A dihasilkan dari pesan asli M dengan rumus $A = (M_i \cdot K) \bmod P$, di mana P adalah bilangan prima besar. Tugas Anda adalah merekonstruksi pesan asli M (dengan $0 \leq M_i < P$) jika diberikan A, K, dan P. Namun, kunci K

tidak selalu memiliki invers modulo, sehingga Anda perlu menentukan apakah dekripsi mungkin dan, jika memiliki invers modulo, maka pesan asli dicari.

Input:

- Baris pertama: tiga bilangan bulat *N, K, P* (*1 ≤ N ≤ 10^5, 1 ≤ K, P≤10^9*, *P* prima).
- Baris kedua: N bilangan bulat A_i

Output:

- Jika dekripsi tidak mungkin (karena *K* tidak memiliki invers modulo *P*), maka cetak "IMPOSSIBLE"
- Jika mungkin, maka cetak *N* bilangan bulat *M* (pesan asli), dipisahkan spasi.

Contoh Input:

```
3 2 5
4 1 3
```

Contoh Output:

```
2 3 4
```

Penjelasan

- N=3, K=2, P=5, array terenkripsi A=[4,1,3]
- Periksa apakah A memiliki invers modulo P: gcd(K,P) = gcd(2,5) = 1, sehingga invers ada.
- Hitung inveers modulo K^{-1} mod P: $2 \cdot 3 \equiv 1 \bmod 5$, jadi $K^{-1} = 3$.
- Dekripsi: M_i = (A_i . K^{-1}) mod P .
  - $M_1 = (4 \cdot 3) \bmod 5 = 12 \bmod 5 = 2$
  - $M_2 = (1 \cdot 3) \bmod 5 = 3$
  - $M_3 = (3 \cdot 3) \bmod 5 = 9 \bmod 5 = 4$
- Pesan asli [2 3 4]

Soal 6: Fibonacci XOR

Diberikan dua bilangan *L* dan *R*, hitung hasil XOR dari semua bilangan Fibonacci dari *F*(*L*) hingga *F*(*R*). Deret Fibonacci dimulai dari (0,1,1,2,3,5,8,…). Indeks selalu diawali dengan 0. XOR merupakan operasi gerbang logika yang dalam sstem bilangan biner akan menghasilkan 1 jika dua operan berbeda nilai (0 dan 1), tapi akan menghasilkan 0 jika dua operan memiliki nilai sama.

Input:
L R, (L adalah indeks awal, sedangkan R adalah indeks akhir)

Output:
XOR dari F(L) hingga F(R)

Contoh Input:

```
3 6
```

Contoh Output:

```
12
```

Penjelasan:
$F(3)=2$, $F(4)=3$, $F(5)=5$, $F(6)=8$, maka hasil XOR adalah: $2 \oplus 3 \oplus 5 \oplus 8 = 12$