AtCoder

AtCoder Beginner Contest 046: *Upsolving*

Prof. Edson Alves - UnB/FGA

2020

Sumário

A - AtCoDeer and Paint Cans

Problema

AtCoDeer the deer recently bought three paint cans. The color of the one he bought two days ago is a, the color of the one he bought yesterday is b, and the color of the one he bought today is c. Here, the color of each paint can is represented by an integer between 1 and 100, inclusive.

Since he is forgetful, he might have bought more than one paint can in the same color. Count the number of different kinds of colors of these paint cans and tell him.

Entrada e saída

Constraints

- $1 \leq a, b, c \leq 100$

Input

Input is given from Standard Input in the following format:

a b c

Output

Print the number of different kinds of colors of the paint cans.

Exemplo de entradas e saídas

Entrada	Saída
3 1 4	3
3 3 33	2

Solução

- O problema consiste em determinar quantos são os números distintos dados na entrada
- Deste modo, a resposta do problema deve ser 1, 2 ou 3
- A resposta será 1 apenas se $a = b = c$
- Se a igualdade acima não é verdadeira, a resposta será igual a 2 se $a = b$ ou $a = c$ ou $b = c$
- Em todos os demais casos a resposta é igual a 3
- É possível, entretanto, utilizar um conjunto (`set` no C++), estrutura de dados que armazena apenas elementos únicos
- Nesta abordagem, a resposta será o tamanho do conjunto após a inserção dos elementos a, b e c

Solução $O(1)$

```cpp
#include <bits/stdc++.h>

using namespace std;

int main()
{
    set<int> s;
    int x;

    for (int i = 0; i < 3; ++i)
    {
        cin >> x;
        s.insert(x);
    }

    cout << s.size() << '\n';

    return 0;
}
```

B - Paiting Balls with AtCoDeer

Problema

There are N balls placed in a row. AtCoDeer the deer is painting each of these in one of the K colors of his paint cans. For aesthetic reasons, any two adjacent balls must be painted in different colors.

Find the number of the possible ways to paint the balls.

Entrada e saída

Constraints

- $1 \leq N \leq 1000$
- $1 \leq K \leq 1000$
- The correct answer is at most $2^{31} - 1$.

Input

Input is given from Standard Input in the following format:

N K

Output

Print the number of the possible ways to paint the balls.

Exemplo de entradas e saídas

Entrada	Saída
2 2	2
1 10	10

Solução

- AtCoDeer pode pintar a primeira bola com qualquer uma das K cores
- Já para a segunda bola restam apenas $K - 1$ opções de escolha, uma vez que não ser escolhida a mesma cor que foi escolhida para a primeira bola
- Para a terceira permanecem $K - 1$ escolhas, pois embora não possa ser selecionada a cor escolhida para a segunda bola, já não há mais impedimento para a cor escolhida para a primeira
- Logo, há K escolhas para a primeira bola e $K - 1$ para as demais, de modo que a resposta S será dada por

$$S = K(K-1)^{N-1}$$

- Portanto a solução tem complexidade $O(N)$ (ou $O(\log N)$, se for utilizado o algoritmo de exponenciação rápida)

Solução $O(N)$

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;

int main()
{
    ll N, K;
    cin >> N >> K;

    ll ans = K;

    for (int i = 1; i < N; ++i)
        ans *= (K - 1);

    cout << ans << '\n';

    return 0;
}
```

C - AtCoDeer and Election Report

Problema

AtCoDeer the deer is seeing a quick report of election results on TV. Two candidates are standing for the election: Takahashi and Aoki. The report shows the ratio of the current numbers of votes the two candidates have obtained, but not the actual numbers of votes. AtCoDeer has checked the report N times, and when he checked it for the i-th ($1 \leq i \leq N$) time, the ratio was $T_i : A_i$. It is known that each candidate had at least one vote when he checked the report for the first time.

Find the minimum possible total number of votes obtained by the two candidates when he checked the report for the N-th time. It can be assumed that the number of votes obtained by each candidate never decreases.

Entrada e saída

Constraints

- $1 \leq N \leq 1000$
- $1 \leq T_i, A_i \leq 1000$ $(1 \leq i \leq N)$
- T_i and A_i $(1 \leq i \leq N)$ are coprime.
- It is guaranteed that the correct answer is at most 10^{18}.

Entrada e saída

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{ll}
N & \\
T_1 & A_1 \\
T_2 & A_2 \\
\vdots & \\
T_N & A_N
\end{array}
$$

Output

Print the minimum possible total number of votes obtained by Takahashi and Aoki when AtCoDeer checked the report for the N-th time.

Exemplo de entradas e saídas

Entrada	Saída
3 2 3 1 1 3 2	10
4 1 1 1 1 1 5 1 100	101

Exemplo de entradas e saídas

Entrada	Saída
5 3 10 48 17 31 199 231 23 3 2	6930

Solução

- Seja X e Y a quantidade de votos para os candidatos 1 e 2, respectivamente
- Como T_i e A_i são coprimos, para que X e Y estejam em proporção $T_i : A_i$ é preciso que $X = kT_i$ e que $Y = kA_i$ para algum k inteiro
- Na primeira apuração, $X = T_1$ e $Y = A_1$ são os menores valores possíveis ($k = 1$)
- Para a i-ésima apuração, determine os menores valores para k_1 e k_2 tais que $X \leq k_1 T_i$ e $Y \leq k_2 A_i$
- Faça $k = \max\{k_1, k_2\}$ e atualize X e Y
- Assim a solução tem complexidade $O(N)$

Solução $O(N)$

```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;
using ii = pair<ll, ll>;

ll solve(int N, const vector<ii>& xs)
{
    ll T1 = xs.front().first, A1 = xs.front().second;

    for (int i = 1; i < N; ++i)
    {
        auto T2 = xs[i].first, A2 = xs[i].second;
        auto k = max((T1 + T2 - 1)/T2, (A1 + A2 - 1)/A2);

        T1 = k*T2;
        A1 = k*A2;
    }

    return T1 + A1;
}
```

Solução $O(N)$

```
22
23 int main()
24 {
25     ios::sync_with_stdio(false);
26
27     int N;
28     cin >> N;
29
30     vector<ii> xs(N);
31
32     for (int i = 0; i < N; ++i)
33         cin >> xs[i].first >> xs[i].second;
34
35     auto ans = solve(N, xs);
36
37     cout << ans << '\n';
38
39     return 0;
40 }
```

D - AtCoDeer and Rock-Paper

Problema

AtCoDeer the deer and his friend TopCoDeer is playing a game. The game consists of N turns. In each turn, each player plays one of the two gestures, Rock and Paper, as in Rock-paper-scissors, under the following condition:

(*) After each turn, (the number of times the player has played Paper) $\leq$ (the number of times the player has played Rock).

Each player's score is calculated by (the number of turns where the player wins) - (the number of turns where the player loses), where the outcome of each turn is determined by the rules of Rock-paper-scissors.

Problema

(*For those who are not familiar with Rock-paper-scissors: If one player plays Rock and the other plays Paper, the latter player will win and the former player will lose. If both players play the same gesture, the round is a tie and neither player will win nor lose.*)

With his supernatural power, AtCoDeer was able to foresee the gesture that TopCoDeer will play in each of the N turns, before the game starts. Plan AtCoDeer's gesture in each turn to maximize AtCoDeer's score.

The gesture that TopCoDeer will play in each turn is given by a string s. If the i-th ($1 \leq i \leq N$) character in s is `'g'`, TopCoDeer will play Rock in the i-th turn. Similarly, if the i-th ($1 \leq i \leq N$) character of s in `'p'`, TopCoDeer will play Paper in the i-th turn.

Entrada e saída

Constraints

- $1 \leq N \leq 10^5$
- $N = |s|$
- Each character in s is `'g'` or `'p'`.
- The gestures represented by s satisfy the condition (*).

Entrada e saída

Input

Input is given from Standard Input in the following format:

s

Output

Print the AtCoDeer's maximum possible score.

Exemplo de entradas e saídas

Entrada	Saída
gpg	0
ggppggggpgg	2

Solução

- Na situação apresentada pelo problema, AtCoDeer deve escolher papel o maior número vezes possível, isto é, $N/2$ vezes
- Uma estratégia para obter o melhor resultado possível seria empatar todos os papéis de seu amigo
- Assim, os $P - p$ papéis restantes seriam vitórias de AtCoDeer
- O mais interessante, porém, é que, uma vez que AtCoDeer tenha $N/2$ papéis, qualquer sequência de jogadas leva ao resultado ótimo
- Assim, a solução tem complexidade $O(N)$

Solução $O(N)$

```cpp
#include <bits/stdc++.h>

using namespace std;

int solve(int N, const string& s)
{
    int p = 0;

    for (auto c : s)
        p += (c == 'p' ? 1 : 0);

    int P = N/2;

    return P - p;
}

```

Solução $O(N)$

```
17 int main()
18 {
19     ios::sync_with_stdio(false);
20
21     string s;
22     cin >> s;
23
24     auto ans = solve((int) s.size(), s);
25
26     cout << ans << '\n';
27
28     return 0;
29 }
```

Referências

1. AtCoder Beginner Contest 046 - Problem A: AtCoDeer and Paint Cans
2. AtCoder Beginner Contest 046 - Problem B: Painting Balls with AtCoDeer
3. AtCoder Beginner Contest 046 - Problem C: AtCoDeer and Election Report
4. AtCoder Beginner Contest 046 - Problem D: AtCoDeer and Rock-Paper