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 - Many Formulas

Problema

You are given a string S consisting of digits between `'1'` and `'9'`, inclusive. You can insert the letter `'+'` into some of the positions (possibly none) between two letters in this string. Here, `'+'` must not occur consecutively after insertion.

All strings that can be obtained in this way can be evaluated as formulas.

Evaluate all possible formulas, and print the sum of the results.

Entrada e saída

Constraints

- $1 \leq |S| \leq 10$
- All letters in S are digits between `'1'` and `'9'`, inclusive.

Input

Input is given from Standard Input in the following format:

```
S
```

Output

Print the sum of the evaluated value over all possible formulas.

Exemplo de entradas e saídas

Entrada	Saída
125	176
9999999999	12656242944

Solução

- Como a string tem, no máximo, 10 caracteres, é possível gerar e avaliar todas as expressões possíveis
- Seja $N = |S|$
- Há $N - 1$ posições onde podem ser inseridos símbolos '+: antes de qualquer caractere s_i, com $i \in [2, N]$:
- Assim, são 2^{N-1} expressões distintas, que podem ser representadas por meio de todos os inteiros de $N - 1$ *bits*, onde um *bit* ligado indica a presença de um '+ na posição indicada
- Cada expressão pode ser avaliada em $O(N)$, logo a solução tem complexidade $O(N2^N)$

Solução $O(N2^N)$

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

using namespace std;
using ll = long long;

ll solve(const string& s)
{
    ll ans = 0;

    for (int p = 0; p < (1 << (s.size() - 1)); ++p)
    {
        ll res = 0, x = 0;

        for (size_t i = 0; i < s.size(); ++i)
        {
            x *= 10;
            x += (s[i] - '0');

            if (p & (1 << i) or i == s.size() - 1)
            {
                res += x;
```

Solução $O(N2^N)$

```
22                  x = 0;
23              }
24          }
25
26          ans += res;
27      }
28
29      return ans;
30  }
31
32  int main()
33  {
34      string s;
35      cin >> s;
36
37      auto ans = solve(s);
38
39      cout << ans << '\n';
40
41      return 0;
42  }
```

D - Snuke's Coloring

Problema

We have a grid with H rows and W columns. At first, all cells were painted white.

Snuke painted N of these cells. The i-th ($1 \leq i \leq N$) cell he painted is the cell at the a_i-th row and b_i-th column.

Compute the following:

- For each integer j ($0 \leq j \leq 9$), how many subrectangles of size 3×3 of the grid contains exactly j black cells, after Snuke painted N cells?

Entrada e saída

Constraints

- $3 \leq H \leq 10^9$
- $3 \leq W \leq 10^9$
- $0 \leq N \leq \min(10^5, H \times W)$
- $1 \leq a_i \leq H \ (1 \leq i \leq N)$
- $1 \leq b_i \leq W \ (1 \leq i \leq N)$
- $(a_i, b_i) \neq (a_j, b_j) \ (i \neq j)$

Entrada e saída

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{ll}
H & W \quad N \\
a_1 & b_1 \\
\vdots & \\
a_N & b_N
\end{array}
$$

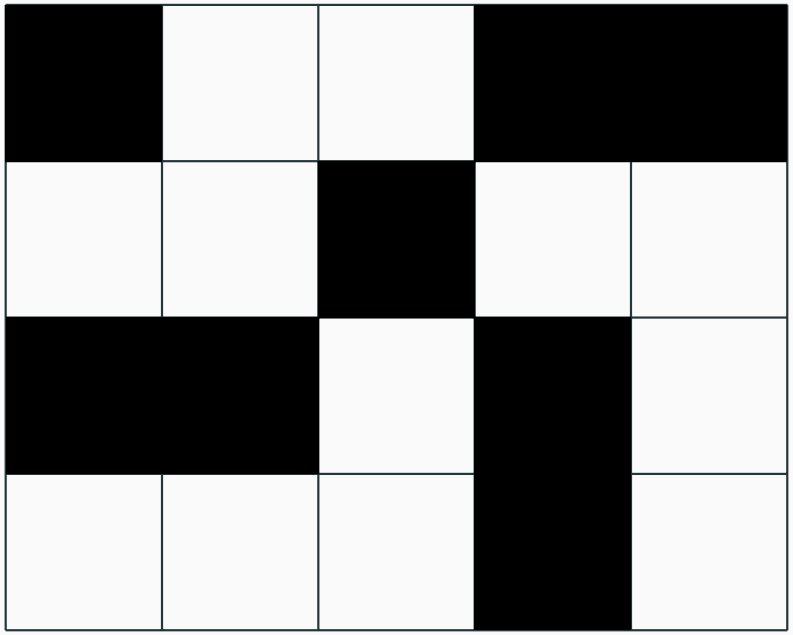
Output

Print 10 lines. The $(j+1)$-th $(0 \leq j \leq 9)$ line should contain the number of the subrectangles of size 3×3 of the grid that contains exactly j black cells.

Exemplo de entradas e saídas

Entrada	Saída
4 5 8	0
1 1	0
1 4	0
1 5	2
2 3	4
3 1	0
3 2	0
3 4	0
4 4	0
	0

Exemplo de entradas e saídas



Solução

- Há um total de $(H-2)(W-2)$ quadrados 3×3 a serem avaliados
- Como $H, W \leq 10^9$, a verificação individual de cada um destes quadrados leva a um veredito TLE
- Contudo, o número total de pontos pintados N é menor ou igual a 10^5
- Cada ponto pertence a um total de 9 destes quadrados
- Assim, é possível resolver o problema verificando, no máximo, 9×10^5 quadrados, pois todos os demais certamente não terão nenhum ponto pintado

Solução

- Considere que cada quadrado seja caracterizado pelo seu canto superior esquerdo
- Cada ponto pintado (x, y) pertence a um dos 9 quadrados cujo canto superior esquerdo tem coordenadas $(x - d_x, y - d_y)$, com $0 \leq d_x, d_y \leq 2$
- Um dicionário deve registrar, para cada possível canto superior esquerdo, quantos pontos pintados fazem parte de seu quadrado associado
- Deve-se atentar que o número de quadrados sem nenhum ponto pintado é a diferença entre o total e os que tem ao menos um ponto pintado
- Esta soluçã tem complexidade $O(N \log N)$

Solução $O(N \log N)$

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

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

vector<ll> solve(ll H, ll W, const vector<ii>& ps)
{
    map<ii, int> hs;

    for (auto p : ps)
    {
        auto x = p.first, y = p.second;

        for (int i = -2; i <= 0; ++i)
        {
            for (int j = -2; j <= 0; ++j)
            {
                auto u = x + i, v = y + j;

```

Solução $O(N \log N)$

```
21                  if (1 <= u and u <= H - 2 and 1 <= v and v <= W - 2)
22                      ++hs[ii(u, v)];
23              }
24          }
25      }
26
27      vector<ll> ans(10, 0);
28
29      for (auto p : hs)
30          ans[p.second]++;
31
32      ans[0] += (H - 2)*(W - 2) - hs.size();
33
34      return ans;
35  }
36
37  int main()
38  {
39      ios::sync_with_stdio(false);
40
```

Solução $O(N \log N)$

```
41      int H, W, N;
42      cin >> H >> W >> N;
43
44      vector<ii> xs(N);
45
46      for (int i = 0; i < N; ++i)
47          cin >> xs[i].first >> xs[i].second;
48
49      auto ans = solve(H, W, xs);
50
51      for (auto x : ans)
52          cout << x << '\n';
53
54      return 0;
55  }
```

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