AtCoder

AtCoder Beginner Contest 043: *Upsolving*

Prof. Edson Alves - UnB/FGA

2020

Sumário

A - Children and Candies (ABC Edit)

Problema

There are N children in AtCoder Kindergarten. Mr. Evi will arrange the children in a line, then give 1 candy to the first child in the line, 2 candies to the second child, ..., N candies to the N-th child. How many candies will be necessary in total?

Entrada e saída

Constraints

- $1 \leq N \leq 100$

Input

Input is given from Standard Input in the following format:

N

Output

Print the necessary number of candies in total.

Exemplo de entradas e saídas

Entrada	Saída
3	6
10	55
1	1

Solução

- O problema consiste em determina a soma

$$S(N) = \sum_{i=1}^{N} = 1 + 2 + \ldots + N$$

- Como o máximo valor de N é relativamente pequeno ($N \leq 100$), é possível computar esta soma por meio de um laço, o que resulta em uma solução $O(N)$
- A soma $S(N)$, porém, pode ser computada por meio da expressão

$$S(N) = \frac{N(N+1)}{2}$$

- Esta expressão pode ser obtida por meio da soma dos termos de uma progressão aritmética de N termos, com $a_1 = 1$ e $a_N = N$
- Usando esta expressão, a complexidade da solução é reduzida para $O(1)$

Solução $O(1)$

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

using namespace std;

int main()
{
    int N;
    cin >> N;

    auto ans = N*(N + 1)/2;

    cout << ans << '\n';

    return 0;
}
```

B – Unhappy Hacking (ABC Edit)

Problema

Sig has built his own keyboard. Designed for ultimate simplicity, this keyboard only has 3 keys on it: the `'0'` key, the `'1'` key and the backspace key.

To begin with, he is using a plain text editor with this keyboard. This editor always displays one string (possibly empty). Just after the editor is launched, this string is empty. When each key on the keyboard is pressed, the following changes occur to the string:

Entrada e saída

- The `'0'` key: a letter `'0'` will be inserted to the right of the string.
- The `'1'` key: a letter `'1'` will be inserted to the right of the string.
- The backspace key: if the string is empty, nothing happens. Otherwise, the rightmost letter of the string is deleted.

Sig has launched the editor, and pressed these keys several times. You are given a string s, which is a record of his keystrokes in order. In this string, the letter `'0'` stands for the `'0'` key, the letter `'1'` stands for the `'1'` key and the letter `'B'` stands for the backspace key. What string is displayed in the editor now?

Entrada e saída

Constraints

- $1 \leq |s| \leq 10$ ($|s|$ denotes the length of s)
- s consists of the letters `'0'`, `'1'` e `'B'`.
- The correct answer is not and empty string.

Input

Input is given from Standard Input in the following format:

```
s
```

Output

Print the string displayed in the editor in the end.

Exemplo de entradas e saídas

Entrada	Saída
01B0	00
0BB1	1

Solução

- A solução do problema consiste em simular o comportamento descrito do editor
- Isto pode ser feito, em C++, por meio de uma variável do tipo string
- No caso dos caracteres `'0'` e `'1'`, o acréscimo à direita corresponde ao método `push_back()`, o qual tem complexidade $O(1)$
- Os caracteres `'B'` correspondem ao método `pop_back()`, que também tem complexidade $O(1)$
- É preciso atentar ao fato de que o método `pop_back()` não deve ser invocado caso a string esteja vazia
- Assim, cada um dos caracteres é processado em $O(1)$, de modo que a solução tem complexidade $O(N)$

Solução $O(N)$

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

using namespace std;

string solve(const string& s)
{
    string res;

    for (auto c : s)
    {
        if (c == '0' or c == '1')
            res.push_back(c);
        else if (not res.empty())
            res.pop_back();
    }

    return res;
}

```

Solução $O(N)$

```
20 int main()
21 {
22     string s;
23     cin >> s;
24
25     auto ans = solve(s);
26
27     cout << ans << '\n';
28
29     return 0;
30 }
```

C - Be Together

Problema

Evi has N integers $a_1, a_2, \ldots, a_N$. His objective is to have N equal integers by transforming some of them.

He may transform each integer at most once. Transforming an integer x into another integer y costs him $(x-y)^2$ dollars. Even if $a_i = a_j$ $(i \neq j)$, he has to pay the cost separately for transforming each of them (See Sample 2).

Find the minimum total cost to achieve his objective.

Entrada e saída

Constraints

- $1 \leq N \leq 100$
- $-100 \leq a_i \leq 100$

Input

Input is given from Standard Input in the following format:

N
$a_1 \quad a_2 \quad \ldots \quad a_N$

Output

Print the minimum total cost to achieve Evi's objective.

Exemplo de entradas e saídas

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

Solução

- Considere, sem perda de generalidade, que a_1 seja o menor dentre todos os números, e que a_N seja o maior dentre eles
- Seja $C(x)$ o custo para transformar todos os elementos $a_1, a_2, \ldots, a_N$
- Observe que $C(x) > C(a_1)$, se $x < a_1$
- Isto porque, estando x à esquerda de todos os números, se aproximar de a_1 reduz todos os custos individuais e, portanto, o custo total da transformação
- De forma análoga, $C(x) > C(a_N)$, se $x > a_N$

Solução

- Logo, o valor de x que minimiza o custo $C(x)$ deve pertencer ao intervalo $[a_1, a_N]$
- Como os limites do problema são relativamente pequenos, computar $C(x)$ para todos os x neste intervalo e registrar o menor valor de $C(x)$ teria complexidade $O(M)$ e veredito AC, onde

$$M = \max\{|a_1|, |a_2|, \ldots, |a_N|\}$$

- Contudo, este problema pode ser resolvido em $O(1)$
- Temos que

$$C(x) = (x - a_1)^2 + (x - a_2)^2 + \ldots + (x - a_N)^2$$

Solução

- Derivando em relação a x obtemos

$$C'(x) = 2(x - a_1) + 2(x - a_2) + \ldots + 2(x - a_N)$$

- A função $C(x)$ terá um ponto crítico em y se $C'(y) = 0$
- Esta igualdade nos dá

$$y = \frac{a_1 + a_2 + \ldots + a_N}{N}$$

- Assim, $C(x)$ será minimizada quando x for a média aritmética dos números $a_1, a_2, \ldots, a_N$
- Como os números devem ser todos inteiros após a transformação, o valor da média deve ser arredondado para o cálculo de $C(y)$, que será a resposta do problema

Solução $O(N)$

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

using namespace std;

int solve(int N, const vector<int>& xs)
{
    int m = (int) round(accumulate(xs.begin(), xs.end(), 0.0) / N);
    int res = 0;

    for (int i = 0; i < N; ++i)
        res += (xs[i] - m)*(xs[i] - m);

    return res;
}

```

Solução $O(N)$

```
16 int main()
17 {
18     int N;
19     cin >> N;
20
21     vector<int> xs(N);
22
23     for (int i = 0; i < N; ++i)
24         cin >> xs[i];
25
26     auto ans = solve(N, xs);
27
28     cout << ans << '\n';
29
30     return 0;
31 }
```

D – Unbalance

Problema

Given a string t, we will call it unbalanced if and only if the length of t is at least 2, and more than half of the letters in t are the same. For example, both `'voodoo'` and `'melee'` are unbalanced, while neither `'noon'` nor `'a'` is.

You are given a string s consisting of lowercase letters. Determine if there exists a (contiguous) substring of s that is unbalanced. If the answer is positive, show a position where such a substring occurs in s.

Entrada e saída

Constraints

- $2 \leq |s| \leq 10^5$
- s consists of lowercase letters.

Partial Score

- 200 points will be awarded for passing the test set satisfying $2 \leq N \leq 100$.

Entrada e saída

Input

Input is given from Standard Input in the following format:

s

Output

If there exists no unbalanced substring of s, print `'-1 -1'`.

If there exists an unbalanced substring of s, let one such substring be $s_a s_{a+1} \ldots s_b$ $(1 \leq a < b \leq |s|)$, and print `'`$a\ b$`'`. If there exists more than one such substring, any of them will be accepted.

Exemplo de entradas e saídas

Entrada	Saída
`needed`	`2 5`
`atcoder`	`-1 -1`

Solução

- Seja N o tamanho da string s
- Há $O(N^2)$ substrings de s, e a verificação de cada substring pode ser feita em $O(N)$, o que levaria a uma solução $O(N^3)$, que receberia veredito TLE, pois $N \leq 10^5$
- É preciso, portanto, encontrar uma forma eficiente de se processar as substrings e também de reduzir o número de substrings a serem verificadas

Solução

- Como qualquer string desbalaceada pode ser dada como resposta do problema, o princípio da casa dos pombos pode reduzir drasticamente o número de substrings a serem verificadas
- Uma string desbalanceada tem mais da metade de seus caracteres repetidos
- Se N é par e s é desbalanceada, ela possui ao menos um par de caracteres consecutivos iguais
- Como uma string com $N = 2$ com dois caracteres iguais é desbalanceada, é suficiente indicar estes caracteres ao invés da string inteira

Solução

- Se N é ímpar, é possível que s seja desbalanceada e que não tenha dois caracteres consecutivos iguais
- Isto só ocorre se o número de repetições é igual a $(N+1)/2$ e o caractere repetido aparece em todos os índices ímpares de s:

$$s = cs_2cs_4 \dots s_{N-1}c$$

- Porém, se $N = 3$ e s é da forma cs_2c, ela também será desbalanceada e poderá ser utilizada como resposta ao invés de s
- Portanto, basta checar apenas estes dois casos, gerando uma solução $O(N)$

Solução $O(|s|)$

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

using namespace std;

pair<int, int> solve(const string& s)
{
    int N = (int) s.size();

    // Caso 1: dois caracteres consecutivos
    for (int i = 1; i < N; ++i)
        if (s[i] == s[i - 1])
            return { i, i + 1 };

    // Caso 2: três caracteres consecutivos, extremos iguais
    for (int i = 2; i < N; ++i)
        if (s[i] == s[i - 2])
            return { i - 1, i + 1 };

    return { -1, -1 };
}

```

Solução $O(|s|)$

```
22 int main()
23 {
24     ios::sync_with_stdio(false);
25 
26     string s;
27     cin >> s;
28 
29     auto ans = solve(s);
30 
31     cout << ans.first << ' ' << ans.second << '\n';
32 
33     return 0;
34 }
```

Referências

1. AtCoder Beginner Contest 043 - Problem A: Children and Candies (ABC Edit)
2. AtCoder Beginner Contest 043 - Problem B: Unhappy Hacking (ABC Edit)
3. AtCoder Beginner Contest 043 - Problem C: Be Together
4. AtCoder Beginner Contest 043 - Problem D: Unbalance