Busca e Ordenação

Algoritmos de busca: problemas resolvidos

Prof. Edson Alves

2020

Faculdade UnB Gama

Sumário

1. AtCoder Beginner Contest 111 – Problem B: AtCoder Beginner Contest 111

2. OJ 10341 – Solve It

3. Codeforces Round #251 (Div. 2) – Problem D: Devu and his Brother

AtCoder Beginner Contest 111 – Problem B: AtCoder Beginner Contest 111

Problema

Kurohashi has never participated in AtCoder Beginner Contest (ABC).

The next ABC to be held is ABC N (the N-th ABC ever held). Kurohashi wants to make his debut in some ABC x such that all the digits of x in base ten are the same.

What is the earliest ABC where Kurohashi can make his debut?

Entrada e saída

Constraints

- $100 \leq N \leq 999$
- N is an integer.

Input

Input is given from Standard Input in the following format:

$$N$$

Output

If the earliest ABC where Kurohashi can make his debut is ABC n, print n.

Exemplo de entradas e saídas

Exemplo de Entrada	Exemplo de Saída
111	111
112	222
750	777

Solução

- Há apenas 9 possibilidades para a resposta: $111, 222, 333, 444, 555, 666, 777, 888$ e 999
- A resposta será o menor dentre entes valores que é maior ou igual a N
- Tal valor pode ser localizado através de uma busca linear simples
- Outra alternativa é utilizar a função `lower_bound()` da STL
- Em ambos casos, a complexidade da solução é constante, pois há, no máximo, 9 comparações a serem feitas, independentemente do valor de N

Solução $O(1)$

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

using namespace std;

int solve(int N)
{
    vector<int> contests;

    for (int d = 1; d <= 9; ++d)
        contests.push_back(100*d + 10*d + d);

    auto it = lower_bound(contests.begin(), contests.end(), N);

    return *it;
}

```

Solução $O(1)$

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

OJ 10341 – Solve It

Problema

Solve the equation:

$$p \times e^{-x} + q \times \sin(x) + r \times \cos(x) + s \times \tan(x) + t \times x^2 + u = 0$$

where $0 \leq x \leq 1$.

Entrada e saída

Input

Input consists of multiple test cases and terminated by an `EOF`. Each test case consists of 6 integers in a single line: p, q, r, s, t and u (where $0 \leq p, r \leq 20$ and $-20 \leq q, s, t \leq 0$). There will be maximum 2100 lines in the input file.

Output

For each set of input, there should be a line containing the value of x, correct up to 4 decimal places, or the string '`No solution`', whichever is applicable.

Exemplo de entradas e saídas

Exemplo de Entrada	Exemplo de Saída
0 0 0 0 -2 1	0.7071
1 0 0 0 -1 2	No solution
1 -1 1 -1 -1 1	0.7554

Solução

- Seja

$$f(x) = p \times e^{-x} + q \times \sin(x) + r \times \cos(x) + s \times \tan(x) + t \times x^2 + u$$

- Observe que $f(x)$ é contínua no intervalo $[0, 1]$
- Assim, caso $f(0)$ e $f(1)$ tenham sinais opostos, há garantias de que existe ao ao menos um $c \in [0, 1]$ tal que $f(c) = 0$
- Logo, se $f(0)$ e $f(1)$ tem sinais iguais, a resposta será '`No solution`'
- Caso contrário, a resposta pode ser determinada por meio de uma busca binária
- A busca deve continuar até que se tenha a garantia de 4 casas decimais corretas
- Estabelecer um limiar $\epsilon = 10^{-6}$ é suficiente para tal precisão

Solução

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

using namespace std;

const double eps { 1e-6 };

double f(double x, int p, int q, int r, int s, int t, int u)
{
    return p*exp(-x) + q*sin(x) + r*cos(x) + s*tan(x) + t*x*x + u;
}

int main()
{
    int p, q, r, s, t, u;

    while (scanf("%d %d %d %d %d %d", &p, &q, &r, &s, &t, &u) > 0)
    {
        auto a = 0.0, b = 1.0;
        auto ya = f(a, p, q, r, s, t, u);
        auto yb = f(b, p, q, r, s, t, u);

```

Solução

```
22          if (ya * yb > 0)
23          {
24              printf("No solution\n");
25              continue;
26          }
27
28          while (fabs(ya - yb) > eps)
29          {
30              auto c = (a + b)/2;
31              auto yc = f(c, p, q, r, s, t, u);
32
33              if (yc * ya > 0)
34              {
35                  a = c;
36                  ya = yc;
37              } else
38              {
39                  b = c;
40                  yb = yc;
41              }
42          }
```

Solução

```
43
44          printf("%.4f\n", a);
45      }
46
47      return 0;
48  }
```

Codeforces Round #251 (Div. 2) – Problem D: Devu and his Brother

Problema

Devu and his brother love each other a lot. As they are super geeks, they only like to play with arrays. They are given two arrays a and b by their father. The array a is given to Devu and b to his brother.

As Devu is really a naughty kid, he wants the minimum value of his array a should be at least as much as the maximum value of his brother's array b.

Now you have to help Devu in achieving this condition. You can perform multiple operations on the arrays. In a single operation, you are allowed to decrease or increase any element of any of the arrays by 1. Note that you are allowed to apply the operation on any index of the array multiple times.

You need to find minimum number of operations required to satisfy Devu's condition so that the brothers can play peacefully without fighting.

Entrada e saída

Input

The first line contains two space-separated integers n, m $(1 \leq n, m \leq 10^5)$. The second line will contain n space-separated integers representing content of the array a $(1 \leq a_i \leq 10^9)$. The third line will contain m space-separated integers representing content of the array b $(1 \leq b_i \leq 10^9)$.

Output

You need to output a single integer representing the minimum number of operations needed to satisfy Devu's condition.

Exemplo de entradas e saídas

Sample Input	Sample Output
2 2 2 3 3 5	3
3 2 1 2 3 3 4	4
3 2 4 5 6 1 2	0

Solução com complexidade $O(N \log N)$

- Seja $f(x) = \sum_i \alpha(a_i, x)$, onde $\alpha(a_i, x) = x - a_i$, se $a_i < x$, e $\alpha(a_i, x) = 0$, caso contrário
- De forma semelhante, seja $g(x) = \sum_j \beta(b_j, x)$, onde $\beta(b_j, x) = b_j - x$, se $b_j > x$, e $\beta(b_j, x) = 0$, caso contrário
- Observe que as funções $f(x)$ e $g(x)$ representam os custos para tornar x uma cota inferior ou superior dos vetores a e b, respectivamente
- Assim, o problema consiste em determinar o mínimo da função $h(x) = f(x) + g(x)$
- Observe que $h'(x) = \mu(a, x) - \rho(b, x)$, onde $\mu(c, x)$ é o número de elementos do vetor a menores do que x e $\rho(b, x)$ é o número de elementos do vetor b maiores do que x
- Como a função $\mu(a, x)$ é não-decrescente e a função $\rho(c, x)$ é não-crescente, a $h'(x)$ é não-decrescente
- Isto significa que $h''(x) \geq 0$, ou seja, $h(x)$ é convexa (unimodal), de modo que seu mínimo pode ser obtido através de uma busca ternária

Solução AC com complexidade $O(N \log N)$

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

using namespace std;
using ll = long long;

const ll oo { 1000000010LL };

ll h(ll x, const vector<ll>& as, const vector<ll>& bs)
{
    ll y = 0;

    for (const auto& a : as)
        y += (a < x ? x - a : 0);

    for (const auto& b : bs)
        y += (b > x ? b - x : 0);

    return y;
}

```

Solução AC com complexidade $O(N \log N)$

```
21 ll solve(const vector<ll>& as, const vector<ll>& bs)
22 {
23     // f(x) é convexa: busca ternária
24     ll a = 0, b = oo, ans = 2000000000000000010LL;
25
26     while (a <= b)
27     {
28         auto m1 = a + (b - a)/3;
29         auto m2 = b - (b - a)/3;
30
31         auto y1 = h(m1, as, bs), y2 = h(m2, as, bs);
32
33         ans = min(ans, y1);
34         ans = min(ans, y2);
35
36         y1 > y2 ? a = m1 + 1 : b = m2 - 1;
37     }
38
39     return ans;
40 }
41
```

Solução AC com complexidade $O(N \log N)$

```
42 int main()
43 {
44     ios::sync_with_stdio(false);
45
46     int N, M;
47     cin >> N >> M;
48
49     vector<ll> as(N), bs(M);
50
51     for (int i = 0; i < N; ++i)
52         cin >> as[i];
53
54     for (int j = 0; j < M; ++j)
55         cin >> bs[j];
56
57     auto ans = solve(as, bs);
58
59     cout << ans << '\n';
60
61     return 0;
62 }
```

Referências

1. AtCoder Beginner Contest 111 – Problem B: AtCoder Beginner Contest 111
2. OJ 10341 – Solve It
3. Codeforces Round #251 (Div. 2) – Problem D: Devu and his Brother