

OJ 13004

At most twice

Prof. Edson Alves – UnB/FGA

Given a positive integer U , find the largest integer L such that $L \leq U$ and L does not contain any digit more than twice.

Input

The input contains several test cases; each test case is formatted as follows. A test case consists of a single line that contains an integer U ($1 \leq U \leq 10^{18}$).

Output

For each test case in the input, output a line with an integer representing the largest number less than or equal to U that does not contain any digit more than twice.

Exemplo de entradas e saídas

Sample Input

2210102960
10000000000000000000
1001223343
20152015

Sample Output

2210099887
998877665544332211
998877665
20152015

- Uma forma de obter a solução para este problema é utilizar o *backtracking*
- O vetor xs será solução se tiver o mesmo número de dígitos de U (talvez com zeros à esquerda) e não tiver um dígito repetido mais do que 2 vezes
- A cada etapa, os candidatos são os dígitos que ainda não se repetiram duas vezes
- A título de poda, os dígitos serão acrescentados à direita, e sempre do maior para o menor candidato disponível
- Assim, a primeira solução encontrada já será a ótima, o que permite encerrar a busca imediatamente

Solução

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string candidates(const string& x)
6 {
7     vector<int> hist(10, 0);
8     string cs;
9
10    for (auto c : x)
11        hist[c - '0']++;
12
13    for (int i = 9; i >= 0; --i)
14        if (hist[i] < 2)
15            cs.push_back(i + '0');
16
17    return cs;
18 }
19
20 string ans;
```

Solução

```
22 void backtracking(string& L, size_t N, const string& U)
23 {
24     if (not ans.empty())
25         return;
26
27     if (L > U)
28         return;
29
30     if (L.size() == N)
31         ans = L;
32
33     auto cs = candidates(L);
34
35     for (auto c : cs)
36     {
37         L.push_back(c);
38         backtracking(L, N, U);
39         L.pop_back();
40     }
41 }
```

Solução

```
43 long long solve(string U)
44 {
45     string xs;
46     ans = "";
47
48     backtracking(xs, U.size(), U);
49
50     return stoll(ans);
51 }
52
53 int main() {
54     ios::sync_with_stdio(false);
55
56     string U;
57
58     while (cin >> U)
59         cout << solve(U) << '\n';
60
61     return 0;
62 }
```