AtCoder Beginner Contest 110

Problema C: String Transformation

Prof. Edson Alves - UnB/FGA

Problema

You are given strings S and T consisting of lowercase English letters.

You can perform the following operation on S any number of times:

Operation: Choose two distinct lowercase English letters c_1 and c_2 , then replace every occurrence of c_1 with c_2 , and every occurrence of c_2 with c_1 .

Determine if S and T can be made equal by performing the operation zero or more times.

1

Entrada e saída

Constraints

- $1 \le |S| \le 2 \times 10^5$
- $\cdot |S| = |T|$
- \cdot S and T consists of lowercase English letters.

Input

Input is given from Standard Input in the following format:

S

T

Output

If S and T can be made equal, print 'Yes'; otherwise, print 'No'.

Exemplo de entradas e saídas

Exemplo de Entrada	Exemplo de Saída
azzel apple	Yes
chokudai redcoder	No
abcdefghijklmnopqrstuvwxyz ibyhqfrekavclxjstdwgpzmonu	Yes

- \cdot Observe que, segundo as regras da operação dada, a correspondência entre dois caracteres de S e T é biunívoca
- \cdot Assim, um caractere c de S deve ser trocado por um caractere d de T que ocorra exatamente o mesmo número de vezes em ambas strings
- · Além do número de ocorrências, é preciso determinar a localização destas ocorrências
- Assim, devem ser mantidos dois dicionários D_S e D_T que mantém o registro das substituições a serem realizadas
- \cdot Assim, para cada caractere c de S, se ainda não tiver sido definida sua substituição e nem a transformação do caractere d de T, ambas substituições são registradas
- \cdot Se uma substituição estiver definida e a outra não, será impossível transformar S em T
- \cdot Se a substituição de c já estiver definida mas for diferente de d, também será impossível obter T a partir de S

```
1#include <bits/stdc++.h>
₃ using namespace std;
5 bool solve(const string& S. const string& T)
6 {
     auto N = S.size();
     map<char, char> s_table, t_table;
8
     for (size t i = 0; i < N; ++i)</pre>
10
          auto c = S[i];
          auto d = T[i]:
          auto it = s_table.find(c);
14
```

```
if (it == s_table.end())
16
              auto jt = t_table.find(d);
18
              if (jt == t_table.end())
                  s_table[c] = d;
                  t table[d] = c:
              } else
24
                  return false;
          } else if (it->second != d)
26
                  return false;
28
      return true:
30
31
```

```
33 int main()
34 {
      ios::sync_with_stdio(false);
35
36
      string S, T;
37
      cin >> S >> T;
38
39
      auto ans = solve(S, T);
40
41
      cout << (ans ? "Yes" : "No") << '\n';
42
43
      return 0;
44
45 }
```