

Problema 10

Dadas dos secuencias de caracteres s_1 y s_2 con longitudes l_1 y l_2 , crea un programa que verifique si es posible permutar los caracteres de s_1 y obtener s_2 .

Solución: Primero verificamos que s_1 y s_2 tengan la misma longitud. De no ser así no existe ninguna permutación. Luego podemos ordenar lexicográficamente ambas cadenas y comparar si el resultado es idéntico. La complejidad de esta solución es $O(n \log n)$ en tiempo donde n es la longitud de ambas cadenas y $O(1)$ en espacio si el ordenamiento no usa espacio adicional.

```
#include<algorithm>
#include<iostream>
using namespace std;
int main () {
    std::string s1;
    std::string s2;
    cin >> s1;
    cin >> s2;
    int ans = 0;
    if (s1.size() != s2.size())
    {
        cout << ans;
    }
    std::sort(s1.begin(), s1.end());
    std::sort(s2.begin(), s2.end());
    if(s1 == s2)
    {
        ans = 1;
    }
    cout << ans;
}
```

Otra solución consiste en contar los caracteres de cada arreglo y verificar que son iguales. Esta solución tiene complejidad de $O(n)$ en expectancia si se usa una tabla hash para el conteo pero usa espacio $O(n)$.