**Immutable String - Problems & Solutions**

[Report Issue](https://github.com/LeetCode-Feedback/LeetCode-Feedback/issues)

You should know whether the string in your favorite language is immutable or not in the previous article. If the string is immutable, it will bring some problems. Hopefully, we will also provide the solution at the end.

*Modification Operation*

Obviously, an immutable string cannot be modified. If you want to modify just one of the characters, you have to create a new string.

*Beware of String Concatenation in Java*

***C++***

*#include <iostream>*

*int main() {*

*string s = "";*

*int n = 10000;*

*for (int i = 0; i < n; i++) {*

*s += "hello";*

*}*

*}*

***Java***

*// "static void main" must be defined in a public class.*

*public class Main {*

*public static void main(String[] args) {*

*String s = "";*

*int n = 10000;*

*for (int i = 0; i < n; i++) {*

*s += "hello";*

*}*

*}*

*}*

You should be very careful with string concatenation. Let's look at an example when we do string concatenation repeatedly in a for loop:

Notice how slow string concatenation is for Java? On the other hand, there is no noticeable performance impact in C++.

In Java, since the string is immutable, concatenation works by first allocating enough space for the new string, copy the contents from the old string and append to the new string.

Therefore, the time complexity in total will be:

*5 + 5 × 2 + 5 × 3 + … + 5 × n  
= 5 × (1 + 2 + 3 + … + n)  
= 5 × n × (n + 1) / 2*,

which is O(n2).

*Solutions*

If you want your string to be mutable, there are some substitutions:

1. **If you did want your string to be mutable, you can convert it to a char array.**

**Java**

// "static void main" must be defined in a public class.

public class Main {

public static void main(String[] args) {

String s = "Hello World";

char[] str = s.toCharArray();

str[5] = ',';

System.out.println(str);

}

}

1. **If you have to concatenate strings often, it will be better to use some other data structures like StringBuilder. The below code runs in O(n) complexity.**

// "static void main" must be defined in a public class.

public class Main {

public static void main(String[] args) {

int n = 10000;

StringBuilder str = new StringBuilder();

for (int i = 0; i < n; i++) {

str.append("hello");

}

String s = str.toString();

}

}