



Java: Írjon olyan függvényt vagy metódust, amely egy karakterláncban vagy sztringben véletlenszerűen összekeveri a karaktereket (véletlenszám-generátor használható)!

```
import java.util.Random;

public class StringShuffle {
    public static String shuffleString(String input) {
        char[] charArray = input.toCharArray();
        Random rand = new Random();
        for (int i = 0; i < charArray.length; i++) {
            int randomIndex = rand.nextInt(charArray.length);
            char temp = charArray[i];
            charArray[i] = charArray[randomIndex];
            charArray[randomIndex] = temp;
        }
        return new String(charArray);
    }
}
```

A fenti kód egy olyan `shuffleString` metódust definiál, amely egy bemeneti sztringet vesz és véletlenszerűen összekeveri a karaktereit. A metódus a sztringet karaktertömbbe helyezi át, majd egy `Random` objektumot használ a karakterek véletlenszerű összekeveréséhez. Végül a kevert karaktertömböt új sztringgé alakítja és visszaadja.



Optimalizáld!

```
import java.util.Random;

public class StringShuffle {
    public static String shuffleString(String input) {
        Random rand = new Random();
        char[] charArray = input.toCharArray();
        for (int i = charArray.length - 1; i > 0; i--) {
            int randomIndex = rand.nextInt(i + 1);
            char temp = charArray[i];
            charArray[i] = charArray[randomIndex];
            charArray[randomIndex] = temp;
        }
        return new String(charArray);
    }
}
```

Az előző megoldásban a kód a karakterek teljes tömbjén végigiterált és minden karakterre egy véletlenszerű indexet választott. Ennek az volt az eredménye, hogy a kód általában többször választott ki ugyanazt az indexet és ezáltal a karaktereket is. Az optimalizált megoldásban a kód az utolsó karaktertől kezdve végigiterál, és minden karakterre egy véletlenszerű indexet választ, amely kisebb mint a jelenlegi index. Így a kód a korábban kiválasztott indexeket nem veszi figyelembe és ezáltal a karakterek összekeverése is gyorsabb lesz.