Java → Implementation of basic algorithms → String algorithms → Hamming distance in Java

Theory: Hamming distance in Java

38 minutes 0 / 5 problems solved

Skip this topic Start practicing

163 users solved this topic. Latest completion was about 18 hours ago.

Current topic:

Hamming distance in Java ···

Topic depends on:

✓ Hamming distance ···

✗ Algorithms in Java ···

Table of contents:

Hamming distance is a simple metric that allows estimating similarity between two strings of equal length. This topic explains how the algorithm for finding the Hamming distance can be implemented in Java.

§1. Implementation in Java

Below is an implementation of a method that allows calculating the Hamming distance for two strings:

```
public static int hammingDistance(String str1, String str2) {
    int hammingDist = 0;

    for (int i = 0; i < str1.length(); i++) {
        if (str1.charAt(i) != str2.charAt(i)) {
            hammingDist += 1;
        }
    }


    return hammingDist;

}
```

The method takes two strings as arguments and returns their Hamming distance (the strings are supposed to be of equal length). The algorithm is pretty straightforward:

1. First, we create a variable to store the Hamming distance and initialize it with zero.
2. Then, using a `for` loop, we iterate through each position and compare corresponding symbols. In case of a mismatch, we increment the value of the variable.
3. After all the symbols are processed, the variable contains the Hamming distance and we return it as a final result.

§2. Usage example

Below you can find an example of how to use the implemented method:

```
int hammingDist = hammingDistance("microscope", "microphone");
System.out.println(hammingDist); // 3
```

Since there are three mismatched symbols in the above strings, the hamming distance between them is equal to three.

§3. Summary

In this topic, we have considered how the algorithm for finding the Hamming distance can be implemented in Java. For now, you should be confident about how to apply this algorithm to estimate the similarity between two strings of equal length. In the following topics, we will consider more sophisticated algorithms that allow for comparing strings of different sizes.

Report a typo

9 users liked this theory. 0 didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Start practicing

Comments (0) Hints (0) Useful links (0) Show discussion