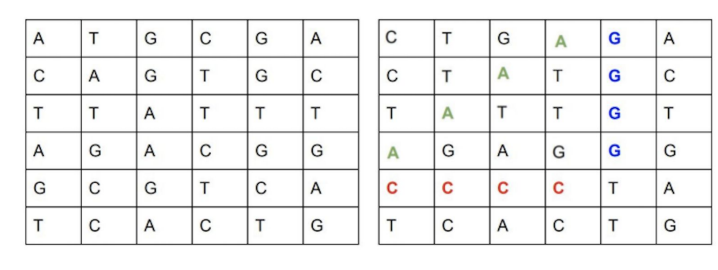
In the distant future, in the chain of evolution, apes and humans are closer and closer. For this reason, it became very difficult to distinguish who is human and who is mutant.

You are a scientist hired to develop a project that detects whether a DNA sequence belongs to a human or a mutant.

For that, you need to develop a program, with a method or function with the following signature (in one of the following languages: (Java / Golang / Javascript (Node) / Python)

boolean isSimian (String[] dna)

You will receive as a parameter an array of Strings that represent each row of a square table of (NxN) with the DNA sequence.



Human / mutant.

The letters of the String can only be: (A, T, C, G), which represents each nitrogenous base of DNA.

You will know if a DNA belongs to a mutant if you find one or more sequences of the same four letters in horizontal, vertical or diagonal directions.

Exemple (Mutant):

String [] dna = {"CTGAGA", "CTGAGC", "TATTGT", "AGAGGG", "CCCCTA", "TCACTG"};

In this case, the call to the isSimian(String[] dna) function should return "true".

Based on this information, develop the algorithm as efficiently as possible according to the challenges below

Level 1:

Develop a method or function that conforms to the proposed signature isSimian(String[] dna), which is able to correctly identify simians.

Level 2:

Create a REST API and host it in some free cloud computing environment (Google App Engine, Amazon AWS, etc).

You must provide a "/simian" endpoint. This service receives a DNA sequence through an HTTP POST with a JSON that contains the following format, for example:

POST → /simian

{

"dna": ["ATGCGA", "CAGTGC", "TTATGT", "AGAAGG", "CCCCTA", "TCACTG"]

}

If the DNA is identified as a mutant, you should return an HTTP 200-OK, otherwise an HTTP 403-FORBIDDEN

Level 3:

Create a database, which stores the DNAs verified by the API. This bank must guarantee uniqueness, that is, only 1 record per DNA.

Provide an extra service "/stats" that responds to an HTTP GET. The answer should be a Json that returns the statistics of DNA checks, where it should inform the amount of mutant DNA's, amount of human DNA's, and the proportion of mutant to the human population.

Here's an example of the answer:

{"count\_mutant\_dna": 40, "count\_human\_dna": 100: "ratio": 0.4}

Comments:

● Keep in mind that we will do a series of tests (For Levels 2 and 3 POSTs and GETs) with

valid and invalid arrays.

● Consider algorithm performance and application response time (For levels 2 and 3), knowing that the API can receive aggressive traffic fluctuations

● The project must contain automated tests, and code coverage must be > 80%.

This challenge was translated from a teste of Mercado Livre.