Turing

1900 Embarcadero Road #104,
Palo Alto, California 94303, U.S.A

# Data Engineering Challenge

## Overview

The purpose of this challenge is to test your ability to obtain large amounts of data from web sources and perform processing in a distributed manner under the given constraints.

In this challenge, you will download 100,000 public Github repositories and perform some processing on the downloaded code. Usage of Amazon Web Services/Google Cloud Platform and multiple instances to perform the processing is highly recommended. AWS free tier lets you use about 750 hours/month of t2.micro instance which should be sufficient for this challenge. Let us know if you face any problems while setting things up.

## Instructions

### Obtaining the data

A list of 100,000 repositories is provided here. These repositories are around 2Mb and the primary language is python. You need to clone all the repositories (only the master branch) and find a fast and cost efficient way of obtaining the data and perform the processing described below.

### Processing

For each repository, our goal is to compute certa

in statistics **only for the Python code** present. Here is the list of items that you need to compute for each repository:

1. **Number of lines of code** [this excludes comments, whitespaces, blank lines].

2. List of **external libraries/packages** used.
3. The **Nesting factor for the repository**: the Nesting factor is the average depth of a nested for loop throughout the code.

```
#Loop 1
for i in range(100):
      for elem in elements:
            ...do something..
            for k in elem:
                  ..do something..

#Loop 2
for i in range(100):
      for elem in elements:
            ..do something..
      for k in range(100):
            ..do something..

#Loop 1 has nesting depth of 3
#Loop 2 has nesting depth of 2
#The average nesting depth for the code is (3+2)/2 = 2.5
```

Note: You must report the average nesting factor for the **entire repository** and not individual files.

4. Code duplication: What percentage of the code is duplicated per file. If the same 4 consecutive lines of code (disregarding blank lines, comments, etc. other non code items) appear in multiple places in a file, all the occurrences except the first occurence are considered to be duplicates.
5. Average number of parameters per function definition in the repository.
6. Average Number of variables defined per line of code in the repository.

## Deliverables

1. An output file '**results.json**' with the results of the computation in the following JSON format. Each item in the Array list represents the result for a single repository.

```
[
{     'repository_url': 'https://github.com/tensorflow/tensorflow',
      'number of lines': 59234,
      'libraries': ['tensorflow','numpy',..],
      'nesting factor': 1.457845,
      'code duplication': 23.78955,
      'average parameters': 3.456367,
      'average variables': 0.03674
}, ......
]
```

2. **The code accompanied with a Readme** containing instructions to run the code. Please mention the dependencies, external packages, etc used in order to execute the code.

**Please upload your code and results on Github as a private repository, invite TuringCom as a collaborator, and email the link to us confirming the submission.**

## Grading Criteria

1. **Use of distributed systems**: We expect you to use multiple nano/micro instances to distribute the workload.
2. **Efficiency**: Since these tasks require you to rummage through a lot of text data, you need to make sure the algorithms and methods used to calculate the statistics are efficient (time and memory).
3. **Accuracy**: How accurate are the statistics? Are all the edge cases covered? We are not looking for exact answers and will accept anything as long as it is within 5% of the original answer.
4. **Optimisations**: Given that we aren't looking for 100% accuracy, can you trade off some accuracy for a much faster method?
5. **Comments**: Well documented and commented code with docstrings, comments and/or a readme.

## Additional Questions

For any additional questions, please use our Github issue tracker page to look at other frequently asked questions or ask your question.