CANADIAN UAVS

Technical Challenge for the Role of Software Developer Intern

This is the second stage of the interviewing process for the position of software developer intern. This technical challenge was designed to showcase a typical problem that a software developer at Canadian UAVs is expected to tackle. Successful candidates for this stage might be invited back to our office.

Preface

You are given the output of two sensors that detect flying objects (airplanes, helicopters, big gliders, etc.). The output consists of coordinates of a detected flying object and the related sensor-specific ID for that object. One sensor output the data as in CSV format and the other in JSON format. While both sensors are excellent and flying objects usually appear on both, it is possible that a flying object will only show up on one list and not the other. Your task is to associate the two sensor outputs taken at the same timestamp and create a consolidated output from them. Both sensor IDs are defined to be non-negative integers only. Coordinates are given as Latitude/Longitude pairs using the World Geodetic System version 84 (WGS 84). You can assume that the maximum disagreement between the sensors should be **100 meters**, after which two reported flying objects are considered separate entities.

Program Input

Each problem "instance" (of which you will have a few) will consist of two input files representing each of the sensor's outputs. You will have one sensor output in CSV and another sensor output in JSON. These sensor outputs are the input files for your program. It should accept two file names (as hardcoded constants at the top of the program or as command line variables).

Program Output

The program should create a single file as an output, which should be called "output.txt". This file will contain the association that your program has created in the following format <CSV_SENSOR_ID>:<JSON_SENSOR_ID>. If a flying object was only picked up by one of the sensors, the other sensor's ID should be reported as -1.

Deliverables

To be considered for the next round of interviews, please view the attached files. There are 2 example input-output pairs named "easy.zip" and "easy2.zip", to which we have provided appropriate solution. Use these to test your program and to understand the task at hand if needed.

You will need to submit the following to eric.eidelberg@canadianuavs.ca by Monday, the **3rd of January at 9:00AM**:

1. Your program's source code as a zip file, together with a readme file that explains what software stack you've used and how we should run it.
2. Solution files for each of the problem instances provided.

You are free to use any programming stack that is reasonable to install. The entire solution should be of your own making, but you are welcome to use any sort of libraries you'd like. **Make sure you clearly cite it in your readme file.** Please don't hesitate to contact Eric at the above email if you have any questions or concerns.