

Assignment

Hi from Digantara,

This is the technical assessment for the Backend Developer (Python) position that you've applied for. Please follow the instructions as accompanied by the question. Best of luck!

Welcome to the world of space and satellites, where we track satellite positions using TLEs (https://en.wikipedia.org/wiki/Two-line_element_set). We have attached a file containing information of about 30 satellites in TLE format - [sample file - 30sats.txt]. In your assignment you need to create a program that do below tasks as explained.

1. Get Satellite location

To find the location of a satellite in space, you can use the sgp4 propagator and the <https://pypi.org/project/sgp4/> library. Simply go through the documentation to understand how to use it.

This library contains functions to retrieve the location of a satellite using TLE data. We have attached a text file containing TLEs for you to use.

Retrieve the position of each satellite for one day using a one-minute time interval. Obtain the data in the following format: time, L(x), L(y), L(z), V(x), V(y), V(z), where L represents location and V represents velocity.

2. Convert data to lat long alt format

```
def ecef2lla(i, pos_x, pos_y, pos_z):
    ecef = pyproj.Proj(proj="geocent", ellps="WGS84", datum="WGS84")
    lla = pyproj.Proj(proj="latlong", ellps="WGS84", datum="WGS84")
    lona, lata, alta = pyproj.transform(
        ecef, lla, pos_x[i], pos_y[i], pos_z[i], radians=False
    )
    return lona, lata, alta
```

NOTE: pyproj is also a library you will be using only in this function.

The above function takes lists Lx, Ly, and Lz as input in list format and returns longitude, latitude, and altitude.

Let's call this result A:

3. Find when it is going from over certain lat long region.

Now that you have converted the location from XYZ to latitude, longitude, and altitude, ask the user for four coordinates (rectangle).

Filter the results from A to only include outputs that are located between user-defined locations.

for example :

Latitude: 16.66673, Longitude: 103.58196

Latitude: 69.74973, Longitude: -120.64459

Latitude: -21.09096, Longitude: -119.71009

Latitude: -31.32309, Longitude: -147.79778

4. Optimize code to reduce computation time

- This program has to be scaled to ingest about 30000 satellites [sample file - 30000sats.txt]
- Optimize performance of the code: RAM, CPU, computation time and so on.
- Use distributed computing to exponentially reduce the computation time.
- Modularized production ready code.

What we are looking for

- We will primarily be looking at your performance in section 4 (optimizing for resources & time).
- Reducing computation time by orders of magnitude is vital.
- You need to be able to use every trick in the book to optimize similar type of codes.
- good documentation is a plus.

Submission

- Share your solution as a git repository with a readme file.
- One pager explaining how distributed computing reduced computation time.

Attachments

[30sats.txt](#)

[30000sats.txt](#)