CSCI 321 – Assignment 3
Winter 2021

Instructions

This assignment is to be completed individually and is due on **Wednesday, March 18, at 11:59 pm**. If you use a late day for this assignment, make sure to log it on Canvas in the Google form link provided. All submissions should be made on Canvas. For this assignment put all your answers in a folder named as follows: **assignment_3_<W&L_ID>.py** (e.g. assignment_3_17xxxxx.py). You are required to only put your single python file **`resolve.py`**. Create an archive of the folder when you submit.

Getting Setup

Most of what you'll need in this assignment is included in the **`resolve.py`** skeleton file included in this archive. You'll need to install the dnspython (https://www.dnspython.org/). You can do so using pip (https://pip.pypa.io/en/stable). In most cases you can do so by running one of the following commands on your system:

- **`pip3 install dnspython`**
- **`pip install dnspython`**
- **`python -m pip install dnspython`**

An essential piece of documentation that you'll need to reference for this assignment is the documentation for dnspython which is here: https://dnspython.readthedocs.io/en/stable/

The Assignment

You'll find a **`resolve.py`** file in your repo. This file contains code that approximates the functionality of the host program. The program takes a domain name, and returns a summary of DNS information about the domain. For example, given the domain "**`yahoo.com`**", host returns the "**A**", "**AAAA**" and "**MX**" records for the domain.

You can test this out by running the resolve.py command as so:

```
python resolve.py yahoo.com.
```

`resolve.py` currently queries a DNS server (**`8.8.8.8`**) to find information about domains. That DNS server handles the recursive part of DNS for you (i.e. **`8.8.8.8`** by default doesn't know where to find **`yahoo.com.`**, so it asks a root server, and the root server tells **`8.8.8.8`** where to find information about .com, so then **`8.8.8.8`** asks the new server where to find **`yahoo.com.`**, etc.)

Your task in this assignment is to implement this recursive functionality on your own. When your version of **`resolve.py`** is run, it will not query **`8.8.8.8`** (or any other recursive resolver). Your **`resolve.py`** will query a root server itself, as well as performing any further needed queries.

The IP addresses of the root servers are hard coded into the `resolve.py` in the global `ROOT_SERVERS` list. Your program should start by querying one of these servers. A very good first step in your solution will be to find where `8.8.8.8` is in the code currently, and make sure you instead query one of the root servers.

Using the responses from what you get from the root severs, you will then ask one of the TLD servers, and so on and so on. The base case of your recursive function will be when you have received the actual **A**, **AAA**, and **MX** records for a given domain.

Additional Implementation Instructions

- Your y a timeout value of 3 seconds for all queries. Any request taking more than 3 seconds to respond should be treated as non-responsive.
- Like the demo in for DNS in class, you can pick a server randomly from the list of servers.
- You should exhaustively try all available servers when trying to answer a query. For example, if a request to a root server gives you 13 servers for the "**.com**" TLD, you should try each of those 13 servers before giving up
- If you receive an error or non-response from a server, you should not retry the server.
- Only query servers over IPv4. You should not query servers over IPv6 when trying to resolve domains.