Prof. KC Claffy
UC San Diego

Assignment 2: Network Ownership and Interconnection (8pts)

Due on Friday, February 3rd 11:59pm

1 Introduction

The goal of this assignment is to understand the business relationships between Internet organizations in the routing context. The business relationship between 2 networks can have great influence on the topology between them. For example, if a large transit provider network have thousands of customers, its connectivity will be significantly more complex than smaller networks with few customers. Any routing changes made by the large transit network will thus have greater impact compared to smaller networks.

In this assignment, you will become familiar with some Internet datasets provided by CAIDA and use such datasets to explore such business relationships. Specifically, you will explore the following two datasets:

- CAIDA AS Organizations Dataset
- CAIDA AS Relationships Dataset

Using the above datasets, you will study various aspects of Internet organizations, including the number of ASes they own and number of customers they have. More details are provided in the sections below.

1.1 Due Date

Friday, February 3rd 11:59pm

1.2 Submission Instructions

There will be 4 tasks in this assignment, and you will complete each task in a separate `python` file. In total you will submit 4 files on Gradescope.: `pa2q1.py`, `pa2q2.py`, `pa2q3.py`, `pa2q4.py`. Your scores will be immediately available upon submission and you have unlimited submission attempts until due date.

2 AS Organizations (4pts)

2.1 Background

CAIDA uses WHOIS information available from Regional and National Internet Registries to infer a mapping from AS numbers to the organizational entities that operate them. More details can be found in the lecture and reading materials of week 3. In this section you will learn to parse the CAIDA AS Organizations dataset and analyze the parsed result.

2.2 Data Access

The CAIDA AS Organizations data can be accessed at: `https://www.caida.org/catalog/datasets/as-organizations/`

In the webpage from the link above, click `Download AS Organizations Dataset` and fill out the access request form. Then you will be redirected the data directory page with a list of AS Organization data files. You will download `20221001.as-org2info.txt.gz` to complete the tasks in Section 2. For convenience, the file is also provided here: `https://cseweb.ucsd.edu/classes/wi23/cse291-e/pa2/20221001.as-org2info.txt.gz`

2.3 Data Format

The as2org files contain two different types of entries: the *AS number* entry and the *organization* entry. An *AS number* entry maps an AS number to an `org_id`, which can later be mapped to an *organization*. An example of the *AS number* entry is shown below:

```
# format: aut|changed|aut_name|org_id|opaque_id|source
1|20120224|LVLT-1|LVLT-ARIN|e5e3b9c13678dfc483fb1f819d70883c_ARIN|ARIN
```

An *organization* entry maps an `org_id` to the organization name. An example of the *organization* entry is shown below:

```
# format: org_id|changed|name|country|source
LVLT-ARIN|20120130|Level 3 Communications, Inc.|US|ARIN
```

Explanation of field names. Fields relevant to this assignment is in bold:
`aut`: **AS number**
`source`: **RIR region the AS number/organization is registered in**
`name`: **Organization name**
`changed`: Date when the entry is last modified
`aut_name`: Name of the AS
`org_id`: Organization ID
`opaque_id`: ID of this entry, generated by the RIR
`country`: Country where the organization is registered

2.4 Data Processing

To parse an AS Organizations file and obtain an AS number to organization name mapping, you will first need to do the following:

- Parse the *AS number* entry to map an AS number to its `org_id`.
- Parse the *organization* entry to map an `org_id` to its company name.

2.5 Task 1: Parse the AS Organizations Dataset (3pts)

Your first task is to parse the AS Organizations dataset and provide a list of AS numbers for any given organization. Your code is expected to do the following:

(1) Read an input file `input.txt`, which contains a **single** organization name. The autograder will place the input file in the same directory it places your `python` file. Example `input.txt`:

```
Massachusetts Institute of Technology
```

(2) Find all AS numbers owned by the organization.

(3) Print the AS numbers, each separated by a newline, into `output.txt`. The ordering of AS numbers does not matter. The autograder expects the output file to be in the same directory as your `python` file. Example `output.txt`:

```
3
63
395326
40
396527
```

You will complete this task in `pa2q1.py`. Your code should be run by running the following command:

```
python3 pa2q1.py
```

2.6 Task 2: Find the Organizations that own at least 20 AS numbers (1pt)

Your second task is to find all organizations that owns ≥ 20 AS numbers. You will complete this task in `pa2q2.py`. Your code should produce an output file `output.txt` that contains those **organization names**, delimited by newlines. You can reuse most of your code from Task 1. Example `output.txt`:

```
Organization ABC
Level 3
A Very Large Network
Another Very Large Network
```

You will complete this task in `pa2q2.py`. Your code should be run by running the following command:

```
python3 pa2q2.py
```

3 AS Relationships (4pts)

3.1 Background

CAIDA provides two AS Relationship datasets: serial-1 and serial-2. The 'serial-1' directory contains AS relationships inferred from BGP using the method described in "AS Relationships, Customer Cones, and Validation". Serial-2 adds links inferred from BGP communities using the method described in "Inferring Multilateral Peering" and traceroute. In this assignment, you will only use data from the 'serial-1' directory. In this section, you will use the CAIDA AS Relationships dataset to identify ASes with a large number of customers.

3.2 Data Access

The CAIDA AS Relationship data can be accessed at: `https://www.caida.org/catalog/datasets/as-relationships/#H2135`

In the webpage from the link above, click `Download AS Relationship Dataset` and fill out the access request form. Then you will be redirected the data directory page with a list of directories. Click on the `serial-1` directory and then you will see a list of AS Organization data files. You will download `20221001.as-rel.txt.bz2` to complete the tasks in the section. For convenience, the file is also provided here: `https://cseweb.ucsd.edu/classes/wi23/cse291-e/pa2/20221001.as-rel.txt.bz2`

3.3 Data Format

The as-rel files contain entries that denotes either a *peer to peer* relationship or a *provider to customer* relationship. If `AS1` is a peer of `AS2`, it is denoted as below:

```
AS1|AS2|0
```

`AS1` is a provider of `AS2`, it is denoted as below:

```
AS1|AS2|-1
```

3.4 Data Processing

You will need to maintain a list of customers, peers, and providers for each AS number. Remember an AS relationship entry is bi-directional: if `AS1` is a customer of `AS2`, then `AS2` is a provider of `AS1`.

3.5 Task 3: Parse the AS Relationship Dataset (3pts)

Your first task is to parse the AS Relationship dataset and provide a list of customer ASes for any given AS. Your code is expected to do the following:

(1) Read an input file `input.txt`, which contains a **single** AS number. The autograder will place the input file in the same directory it places your `python` file. Example `input.txt`:

```
195
```

(2) Find all customers of the given AS.

(3) Print the customer ASes, each separated by a newline, into `output.txt`. The ordering of ASes does not matter. The autograder expects the output file to be in the same directory as your `python` file. Example `output.txt`:

```
12148
2641
46985
22213
1909
```

You will complete this task in `pa2q3.py`. Your code should be run by running the following command:

```
python3 pa2q3.py
```

3.6 Task 4: Find the ASes that have at least 20 customers (1pt)

Your second task is to find all ASes that have ≥ 20 customers. You will complete this task in `pa2q4.py`. Your code should produce an output file `output.txt` that contains those ASes, delimited by newlines. You can reuse most of your code from Task 3. Example `output.txt`:

```
3356
174
7015
7018
```

You will complete this task in `pa2q4.py`. Your code should be run by running the following command:

```
python3 pa2q4.py
```