

Deadlines:

For all Groups: December 6 2021

Grading system:

For every problem you solve you get a score. Your score is your mark for this laboratory.

- 1 - 1
- 2 - 1.5
- 3.1 - 0.5
- 3.2 - 0.5
- 3.3 - 2
- 3.4 - 0.5
- 3.5 - 0.5
- 3.6 - 1
- 4.1 - 0.5
- 4.2 - 1
- 4.3 - 1

1. Random vulnerability

You've stumbled onto a significant vulnerability in a commonly used cryptographic library. It turns out that the random number generator it uses frequently produces the same primes when it is generating keys.

Exploit this knowledge to factor the (hexadecimal) keys below, and enter your answer as the last six digits of the largest factor you find (in decimal).

Key 1:

```
1c7bb1ae67670f7e6769b515c174414278e16c27e95b43a789099a1c7d55c717b2f0a0442a7d4
9503ee09552588ed9bb6eda4af738a02fb31576d78ff72b2499b347e49fef1028182f158182a0b
a504902996ea161311fe62b86e6ccb02a9307d932f7fa94cde410619927677f94c571ea39c7f41
05fae00415dd7d
```

Key 2:

```
2710e45014ed7d2550aac9887cc18b6858b978c2409e86f80bad4b59ebcbd90ed18790fc56f53f
fab0e4a021da2e906072404a8b3c5555f64f279a21ebb60655e4d61f4a18be9ad389d8ff05b99
4bb4c194d8803537ac6cd9f708e0dd12d1857554e41c9cbef98f61c5751b796e5b37d338f5d9b
3ec3202b37a32f
```

2. RSA

Make a program that will implement the RSA algorithm made by you and with help of this program you can encrypt any string and then decrypt it.

Rule:

You can't use libraries, just remember the algorithm that you study and implement it.

3. Matrix

You have a set of 20 people connected via a friendship matrix. The whole list is given in matrix.txt..

3.1 Friends

Find the person with the most friends.

3.2 Sort

Sort all the people by the number of friends.

3.3 Let's do ratings

How to do that? Well, each person in the graph is connected to everyone else at some level. Therefore, each person will have a list of connections which is as long as the total list of people in the graph (in our case, 20). You then have to compute the *shortest path* from each of the nodes to each of the other nodes.

For example, let's say that you found that from node 0 you can reach node 3 in 5 steps (that is, the shortest path connecting nodes 0 and 3 has 5 steps). That means that node 3 will be a connection of level 5 to node 0 and will therefore contribute to 0 with 4 points.

As a procedure, you can take each item n and then compute the distances between n and all the other vertices of the graph. You can use these distances to compute the value that is added by each of the other $n - 1$ vertices to n . Sum it and you'll have the value of vertex n .

In order to find the shortest path between two vertices, you'll have to use Dijkstra's *algorithm*. You can find plenty of implementations of that algorithm online.

Compute the points for each person in our network. Let's call it "Rating"

3.4 Influential people

Let's say that each of these people has a certain rate of posting content. Obviously, people who communicate more are much more influential. Suppose that you need to promote a new brand using social media. We found out how often each of these 20 people writes something on their walls. You can find it in `influence.txt`

Whom of these people will you contact? Why? Be advised that not only the frequency of posting matters, but also the number of friends!

Use the data from the previous exercise and find the new "Rating" for each person by multiplying it with 0.5 of the posting rate.

Please sort the people by the newly computed rating.

3.5 Analyze your content

You are publishing a book and would like to promote it through the use of social media. The book's title is "From T-Rex to Multi Universes: How the Internet has Changed Politics, Art and Cute Cats." You have done some research in the world's most popular social network and have found that the range of interests is stored in `interests.txt`

Analyze your title and see what specter of interests is your book marketable to.

3.6 Promote it

We have provided you with a list of interests of each of these people. You can find it in `interests.txt`.

Considering the set of interests you have chosen, who of them would you market the book to? Let's say that a person has 5 of her interests coinciding with your books and she has a Rating of 346. Multiply her rating with the $0.2 * \text{coinciding interests}$ to see a final score. Sort the people by this final score.

Provide us with a list of 5 people we should contact to make your book a bestseller! Please use the names found in `people_interests.txt`.

4. Network

The dataset

The dataset is a text file where every line represents a JSON object that describes a tweet (tweet.txt). It was fetched using twitter stream API, hence we're dealing with real life data (yay).

4.1 Popular Hashtag

Write a program that prints on the screen 10 most popular #hashtags followed by the number of occurrences of the #hashtag.

4.2 Tokenizer

Let's do some emotional analyses.

In this file AFINN-111.txt you'll find an emotion dictionary for English words. Every word mentioned in the dictionary is followed by a numerical value in the range of -5 to 5. The numerical value describes the word emotional impact where -5 is the most negative and 5 is the most positive.

Your task is to find the emotional value for every tweet. First step would be to extract every word from the tweet body. I recommend using an nltk tokenizer (similar to TPI Lab 3). Then you find out the emotional value for every word (if it has one). You finish by summing the emotion rating.

Write a program that will store the computed result in a text file. Every line should represent the tweet id followed by the computed emotional value.

4.3 Top

Write a program that prints on the screen 10 most positive tweets and 10 most negative tweets.

Bonus Problem

Let's visualize some data

For data visualization we're going to use cytoscape or igraph. Cytoscape is a tool for drawing graphs and various graph manipulations that can help you extract valuable conclusions out of it. You can download it here <https://cytoscape.org/>

Igraph is a library for many programming languages that help with drawing graphs and various graph manipulations that can help you extract valuable conclusions out of it. Link to library for python: <https://igraph.org/python/>

Loading the graphs

A major part of your job is to create a file that can be fed to one of these tools. The more descriptive XML based formats allow a more detailed graph customization. But in our case a CSV (comma separated values) formatted file should suffice. You can google more details about CSV format, once you're done you can find out in documentation how to format your file in order to be able to load it to the respective tool.

4.4 Relations

Firstly you have to select 200 tweets from the initial dataset. In the current dataset are 10K tweets. In order to select your 200 tweets you have to do the following. Compute $\text{value} = 200 * \text{num\curel_meu_din_catalog}$ (from the big list 1..~30) and start reading the tweets from the line number equal to value.

Example: Orice FAFer, $\text{num\curel_meu_catalog} = 23$, $\text{value} = 23 * 200 = 4600$. So I select my 200 tweets from the dataset starting the line 4600.

From every tweet message text, you have to extract the words (you guessed right, using a tokenizer). Every word should represent a graph node. The graph edges represent the connection between words in every tweet message.

Example: tweet 1: "What a sunny day", tweet 2: "This day is awful"

The words have to be connected in the following way:

What -> a, sunny, day

a -> What, sunny, day

day -> What, a, sunny, This, is, awful

This -> day, is, awful

is -> This, day, awful

awful -> This, day, is

First step

Write a program that will generate a CSV file that represents the word connection in the selected 200 tweets.

Second step

Load the generated csv in the tool that you selected. Set a different node size depending on the number of connections it has with other nodes.

4.5 Filtering

Let's clean the data a bit. Once you finish exercise 4 you will be able to distinguish the noise in your data. Nodes that have lots of connections like a, t, http, and other irrelevant nodes that do not bring any value, more than that it only pollutes our graph.

Your task is to select a list of handpicked words (of the size at least 15) that you consider to be most irrelevant for your current graph (usually the big unimportant nodes). The second step is to patch your script from exercise 4 such that when you generate the CSV it will not include the nodes from your list of selected words.

Once you've generated the new CSV repeat the **second step** from exercise 4.