Homework 1 (Due: Mar. 17, 2021 (11:59 PM))

Name: Your name

Student ID: Your ID Late submission Days used here: 0 (if there is, please modify here)

[Please SUBMIT (1) YOUR IPYNB AND (2) PDF (please use FILE/DOWNLOAD AS/PDF or PRINT PREVIEW/PRINT AS PDF with "printed output") TO iCampus]

For this homework you cannot use the python library scikit-learn (sklearn). You can use the python package BeautifulSoup to parse web pages.

In this assignment you will retrieve and parse webpages. The text file "urls.txt" contains a list of urls for the webpages to be parsed. Each line in the text file corresponds to a url. Use BeautifulSoup to fetch each webpage.

Note: For all questions, the words should be converted to lower case.

Q1: Part 1 (5)

Parse the first webpage document to retrieve the text enclosed in paragraph tags, find the words that end in "ing", and count how many times each word appears.

Sort these words in decreasing order of frequency, and write the words (along with their corresponding frequencies) in an output file named "Q1_Part1.txt". The most frequent word should appear at the top and the least frequent word at the end, and the format of the output file should be: word TAB frequency

Example:

sorting 10 training 8 broadening 6 extracting 3 evergrowing 2 coming 1

In [2]:
```
### YOUR CODE HERE
a=5
a+5
```

Out[2]: 10

Q1: Part 2 (5)

Stop words are natural language words which have very little meaning, such as "and", "the", "a", "an", and similar words.

Repeat Part 1, but before counting, remove the stop words given in the file "stop_words.txt". The ouput for Part 2 should have the same format as Part 1, and should be written to an output file named "Q1_Part2.txt".

In [ ]:
```
### YOUR CODE HERE
```

Q2 (10)

Again, parse the first webpage document, but this time find and count all outgoing links to other webpages, and write the output to a file named "Q2.txt", with each url on a new line.

The format of the output file should: number of outgoing urls in the first line, followed by each url on a new line. For example, 4
https://eng.skku.edu/eng/edu/education.do (https://eng.skku.edu/eng/edu/education.do)
https://eng.skku.edu/eng/Research/industry/researchStory.do (https://eng.skku.edu/eng/Research/industry/researchStory.do)
https://eng.skku.edu/eng/Univ-Industry/Research-Business-Found/FactsandFigures.do (https://eng.skku.edu/eng/Univ-Industry/Research-Business-Found/FactsandFigures.do) https://eng.skku.edu/eng/CampusLife/support/employment.do (https://eng.skku.edu/eng/CampusLife/support/employment.do)

In [ ]:
```
### YOUR CODE HERE
```

Q3: Part 1 (10)

1. Retrieve and parse multiple web pages. The text file "urls.txt" contains a list of webpages to be parsed. Each line in the text file corresponds to a url. Use BeautifulSoup to fetch each webpage and parse it as specified below.
2. For each webpage document do the following:
   A. Retrieve all text enclosed in paragraph tags.
   B. Convert the text to lowercase.
   C. Strip out punctuation. Note: if you use translate() with string.punctuation, then it may not strip out all characters. Use a regular expression involving \W to strip out all non alpha-numeric characters.
   D. Tokenize into words based on whitespace separation.
3. Find the number of unique words in each webpage document.
4. Find the Length of each webpage document. The length of a document is defined as the total number of words in the document (not just unique words).
5. For each of the following words: “statistics”, “analytics”, “data”, and “science”, a. Find Term Frequency (tf). The term frequency (tf) of a term (word) is defined as the number of times that term t occurs in document d, divided by the total number of words in the document. The tf of a word depends on the document under consideration.

   b. Find Inverse Document Frequency (idf). The inverse document frequency of a word is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that ratio. The idf of a word doesn't depend on any documnet in which the word is present. To calculate the idf, you will have to use the log function. The base for the log function must be e.

   c. Find tf-idf. The tf-idf of a word is the product of the term frequency of the word in document d, and its inverse document frequency. The tf-idf of a word depends on the document under consideration.

   Reference: https://en.wikipedia.org/wiki/Tf%E2%80%93idf (https://en.wikipedia.org/wiki/Tf%E2%80%93idf)

The output should be written to an output file named "Q3_Part1.txt".

The format of the output file is as shown below:

1. Number of unique words in documents: [702, 723, 280]
2. Length of documents: [1711, 1928, 563]
3. tf statistics: [0.0070134424313267095, 0.0025933609958506223, 0.0] analytics: [0.0029222676797194622, 0.0031120331950207467, 0.0] data: [0.056107539450613676, 0.05446058091286307, 0.0] science: [0.03798947983635301, 0.011410788381742738, 0.028419182948490232]
4. idf statistics: 0.510825623766 analytics: 0.510825623766 data: 0.223143551314 science: 0.0
5. tf-idf statistics: [0.0028437061936282715, 0.0010515173965460695, 0.0] analytics: [0.0011848775806784465, 0.0012618208758552834, 0.0] data: [0.022749649549026172, 0.022081865327467459, 0.0] science: [0.0, 0.0, 0.0]

The above values are for the first three webpage urls given in the file "urls.txt". The number of unique words in documents, average length of documents, tf and tf-idf values for the four words, must be in the order of the urls given in "urls.txt".

```
In [1]: ### YOUR CODE HERE
```

Q3: Part 2 (10)

Repeat Part 1, but first remove the stop words given in the file "stop_words.txt".
The ouput for Part 2 should have the same format as Part 1, and should be written to an output file named "Q3_Part2.txt". Note: The length of document, in this case, will not include stop words. Similarly, the number of unique words in documents, and the calculation of tf, idf, tf-idf should be done after removing the stop words.

Sample output for the file "urls.txt"

1. Number of unique words in documents: [596, 600, 231]
2. Length of documents: [1020, 1079, 357]
3. tf statistics: [0.011764705882352941, 0.004633920296570899, 0.0] analytics: [0.004901960784313725, 0.005560704355885079, 0.0] data: [0.09411764705882353, 0.09731232622798888, 0.0] science: [0.06372549019607843, 0.020389249304911955, 0.04481792717086835]
4. idf statistics: 0.405465108108 analytics: 0.405465108108 data: 0.405465108108 science: 0.0
5. tf-idf statistics: [0.0047701777424489925, 0.0018788929940137366, 0.0] analytics: [0.0019875740593537469, 0.002254671592816484, 0.0] data: [0.03816142193959194, 0.039456752874288473, 0.0] science: [0.0, 0.0, 0.0]

The above values are for the first three webpage urls given in the file "urls.txt". The number of unique words in documents, average length of documents, tf and tf-idf values for the four words, must be in the order of the urls given in "urls.txt".

In [ ]:
```
### YOUR CODE HERE
```