Homework 4. Fequent Words and Web scraping

Double Click here to edit this cell

- Name: 전기범
- Student ID: 201703091
- Submission date: 2019/04/18

Problem 1 (15 pts)

- Project Gutenberg is a volunteer effort to digitize and archive cultural works.
- Moby-Dick is an 1851 novel by American writer Herman Melville.
- You can find Moby-Dick in an ordinary text format at https://www.gutenberg.org/files/2701/old/moby10b.txt (https://www.gutenberg.org/files/2701/old/moby10b.txt)
- Use **requests** module to get the text.

- We want to compute word frequency of words appearing in mobydick and generate WordCloud
  - First, you must split the text into words.
  - To do that, **you find word delimiters** (for example, . or , ... whatever).
  - To split into words, use **re** (regular expression module)
  - Numbers should not be words.
  - Null string is not a word.
  - Any delimiters should not be words.
  - (Upper or lower) Cases does not matter in words.

웹사이트의 데이터 긁어오기, 가장 자주 사용되는 단어 찾아보기

1.1 Print top 50 most common words (5 pts)

In [1]:
```
import re, requests
freq={}
'''with open("moby10b.txt", 'r', encoding='utf-8') as f:    # txt 파일 이용시
    #L = re.findall('[a-z]+', f.read().lower())'''

response = requests.get('http://www.gutenberg.org/files/2701/old/moby10b.txt')    # 크롤링 이용시
L = re.findall('[a-z]+', response.text.lower())

for word in L:
    cnt = freq.get(word, 0)
    freq[word] = cnt + 1

print([(key,value) for key,value in sorted(freq.items(), key =lambda x:x[1], reverse=True)][:50])
```

```
[('the', 14512), ('of', 6676), ('and', 6471), ('a', 4774), ('to', 4690), ('in', 4190), ('that', 3095), ('it', 2542), ('his', 2530), ('i', 2128), ('he', 1896), ('but', 1823), ('s', 1811), ('as', 1750), ('is', 1748), ('with', 1729), ('was', 1647), ('for', 1643), ('all', 1537), ('this', 1437), ('at', 1332), ('by', 1232), ('whale', 1228), ('not', 1162), ('from', 1103), ('on', 1077), ('so', 1073), ('him', 1067), ('be', 1058), ('you', 949), ('one', 934), ('there', 870), ('now', 787), ('had', 779), ('have', 773), ('or', 761), ('were', 685), ('they', 669), ('which', 650), ('like', 648), ('me', 634), ('then', 632), ('some', 621), ('what', 620), ('their', 620), ('are', 611), ('when', 608), ('an', 600), ('no', 592), ('my', 589)]
```

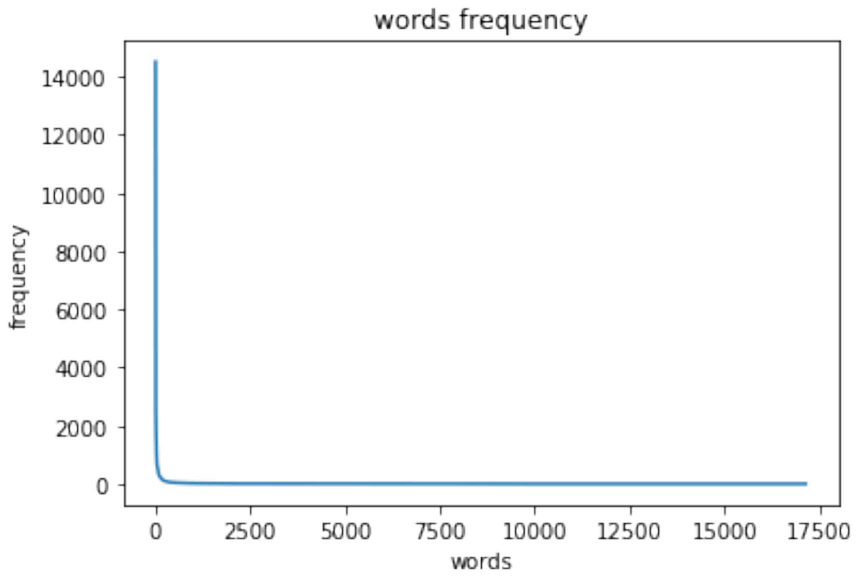
Your output should be like the following:

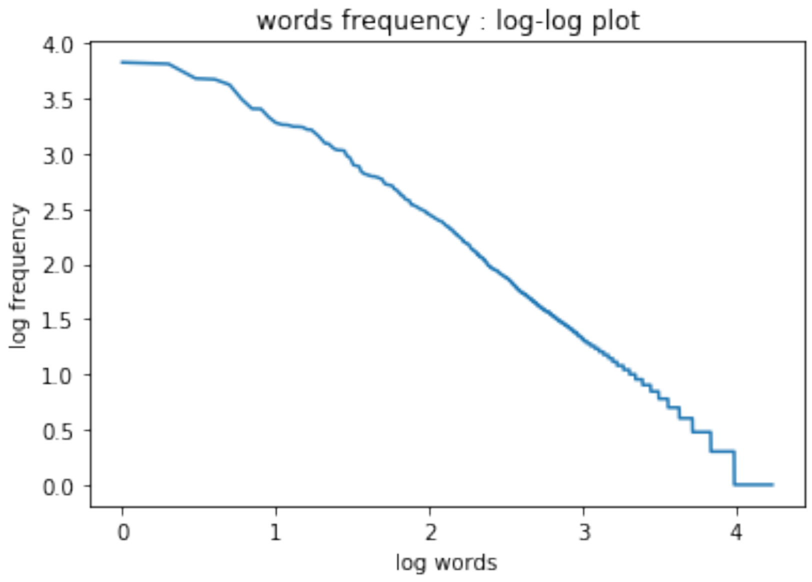
```
[('the', 14512), ('of', 6676), ('and', 6471), ('a', 4774), ('to', 4690), ('in', 4190), ('that', 3095), ('it', 2542), ('his', 2530), ('i', 2128), ('he', 1896), ('but', 1823), ('s', 1811), ('as', 1750), ('is', 1748), ('with', 1729), ('was', 1647), ('for', 1643), ('all', 1537), ('this', 1437), ('at', 1332), ('by', 1232), ('whale', 1228), ('not', 1162), ('from', 1103), ('on', 1077), ('so', 1073), ('him', 1067), ('be', 1058), ('you', 949), ('one', 934), ('there', 870), ('now', 787), ('had', 779), ('have', 773), ('or', 761), ('were', 685), ('they', 669), ('which', 650), ('like', 648), ('me', 634), ('then', 632), ('some', 621), ('what', 620), ('their', 620), ('are', 611), ('when', 608), ('an', 600), ('no', 592), ('my', 589)]
```

1.2 Plot word frequency (5 pts)

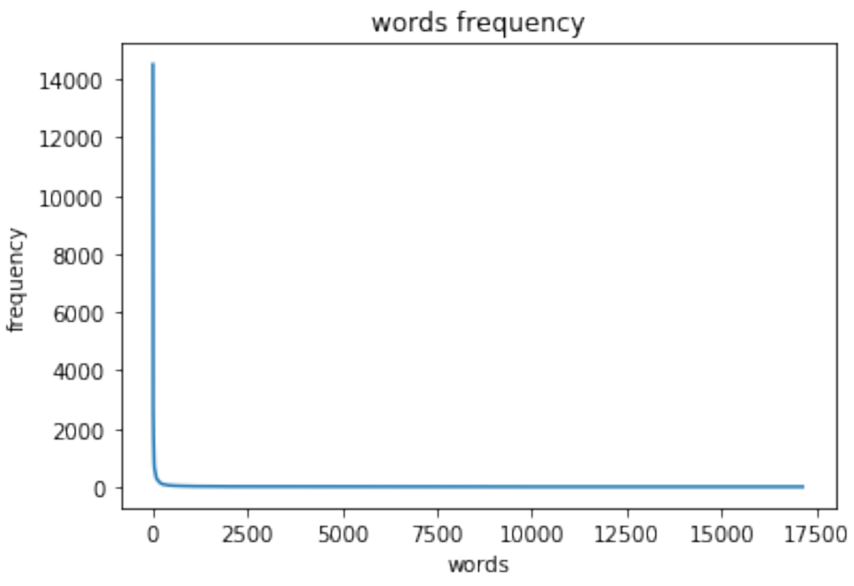
- Sort the word frequency in descending order
- Plot the word frequency
- Plot the word frequency in log-log plot.

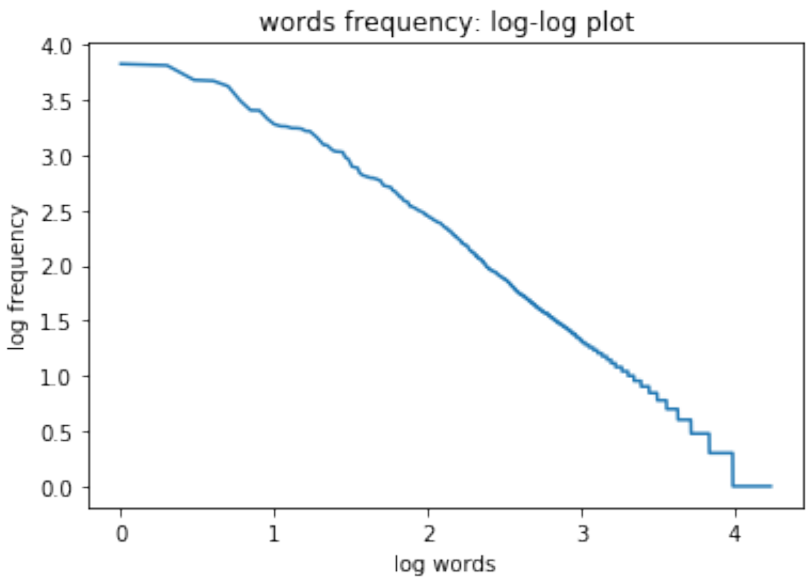
In [2]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

def plot():
    plt.plot(freq_len, word_freq)
    plt.title("words frequency")
    plt.ylabel("frequency")
    plt.xlabel("words")
    plt.show()

def loglog():
    #plt.loglog(freq_len, word_freq)
    np.seterr(divide = 'ignore')
    plt.plot(np.log10(freq_len),np.log10(word_freq))
    plt.title("words frequency : log-log plot")
    plt.ylabel("log frequency")
    plt.xlabel("log words")
    plt.show()

word_freq = np.sort(list(freq.values()))[::-1]
freq_len = np.arange(len(freq))
plot()
loglog()
```


words frequency
frequency
words
0
2000
4000
6000
8000
10000
12000
14000
0
2500
5000
7500
10000
12500
15000
17500


words frequency : log-log plot
log frequency
log words
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
1
2
3
4

Your output be like:





Discussion

- Read this wikipedia article : https://ko.wikipedia.org/wiki/%EC%A7%80%ED%94%84%EC%9D%98_%EB%B2%95%EC%B9%
  (https://ko.wikipedia.org/wiki/%EC%A7%80%ED%94%84%EC%9D%98_%EB%B2%95%EC%B9%
- Discuss what you learned from the distribution.
- WRITE HERE (To edit, double click this cell)

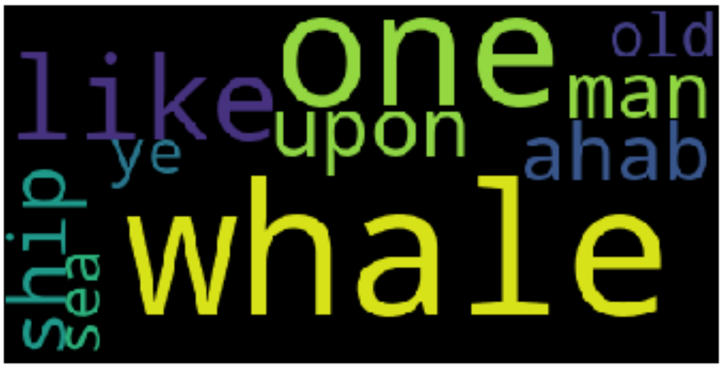
1.3 Word Cloud (5 pts)

- Print top 10 most words except stop words
- Draw word cloud of top 10 most common words

Your output should be like:

```
[('whale', 1228), ('one', 934), ('like', 648), ('upon', 566), ('man', 5
27), ('ship', 518), ('ahab', 511), ('ye', 472), ('sea', 455), ('old', 4
50)]
```

Your output should be like this (but NOT exactly the the same):

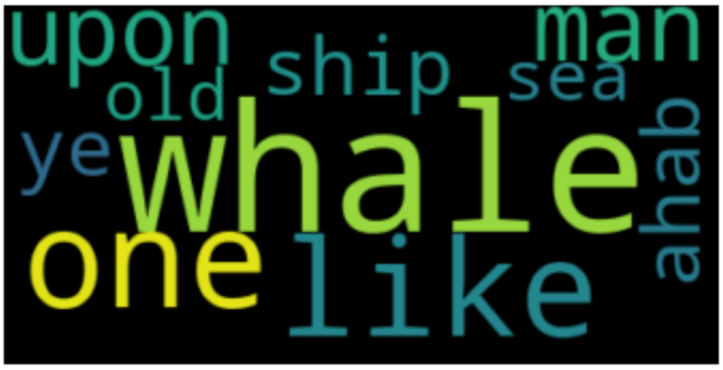


- The following is English stop words list

```
In [3]: stopwords = {'it', 'than', 'out', 'an', 'at', 'until', 'wouldn', 't
oo', 'each', 'off', 'whom', 'not', 'ain', 'weren', "you've", 'isn',
've', "that'll", 'didn', 'couldn', 'with', 'for', 'me', "shouldn't"
, 'those', 'once', 'them', 'him', 'again', 'what', 'to', 's', 'don'
, 'yourselves', "she's", 'd', 'we', 'so', 'does', 'your', 'is', 'su
ch', 'hasn', 'doesn', "doesn't", 'no', 'll', 'their', 'before', 'my
', 'being', 'and', 'but', 'below', 'won', "don't", 't', 'myself', '
very', 'why', "mustn't", 'that', 'been', 'you', "you'd", 'few', 'ot
her', 'ma', 'any', 'having', 'against', 'into', 'on', 'just', 'hers
elf', "hadn't", "mightn't", 'aren', "wouldn't", 'ours', 'about', 't
hen', 'mustn', 'i', 'y', 'should', 'all', 'while', 'himself', 'do',
'up', 'were', 'this', 'most', 'when', 'nor', 'from', 'hadn', 'their
s', 'she', 'be', 'under', 'or', 'will', 'through', 'our', "isn't",
'as', 'between', 'had', 'more', "aren't", "weren't", 'if', 'the', '
am', 'how', 'both', "you're", 'yourself', "couldn't", 'only', 'in',
'itself', 'own', "it's", 'because', 'some', "didn't", 'wasn', 'shan
', "hasn't", 'mightn', 'shouldn', 'here', 'he', 'where', 'm', 're',
'was', 'after', 'has', 'same', "shan't", 'further', "wasn't", 'down
', 'yours', "should've", 'now', "needn't", 'above', 'haven', 'its',
'who', 'of', 'ourselves', 'did', 'these', 'there', 'his', "haven't"
, "won't", 'themselves', "you'll", 'a', 'are', 'which', 'have', 'by
', 'during', 'can', 'hers', 'over', 'her', 'doing', 'o', 'needn', '
they'}
```

In [4]:
```
# YOUR CODE MUST BE HERE
%matplotlib inline
import matplotlib.pyplot as plt
from wordcloud import WordCloud

for stop in stopwords:
    if stop in freq:
        del freq[stop]
words=([(key,value) for key,value in sorted(freq.items(), key =lamb
da x:x[1], reverse=True)][:10])
print(words)

text = ([key for key,value in sorted(freq.items(), key =lambda x:x[
1], reverse=True)][:10])
cloud = WordCloud(max_font_size=100, stopwords=stopwords).generate(
" ".join(text))
plt.figure()
plt.imshow(cloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

```
[('whale', 1228), ('one', 934), ('like', 648), ('upon', 566), ('ma
n', 527), ('ship', 518), ('ahab', 511), ('ye', 472), ('sea', 455),
('old', 450)]
```



Problem 2 (20 pts)

- We want to find how many CS faculty members at CS department of Stanford Univ work on CS research areas.
- First, visit https://cs.stanford.edu/research (https://cs.stanford.edu/research)
- Take a look at the source html of the web page.
- We want to scrape data on all the faculty members
- Run the following two cells and see what happens
- If necessary, install html5lib

```
In [4]: from bs4 import BeautifulSoup
        import requests

        #url = "https://cs.stanford.edu/research"
        url = "https://cs.stanford.edu/research?items_per_page=All&field_fa
        culty_status_value=active"
        soup = BeautifulSoup(requests.get(url).text, 'html5lib')
```

```
In [ ]: print(soup.tbody.prettify())
```

Draw bar charts on research area contributions of Stanford CS faculty

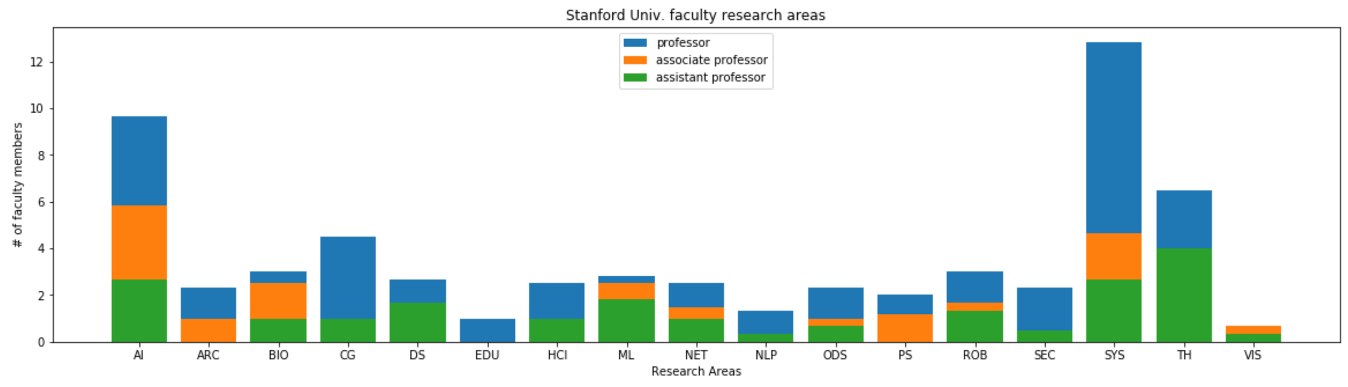
- The followings are research fields

```
['Architecture', 'Artificial Intelligence', 'Computational Biology',
'Computer Graphics', 'Computer Security', 'Computer Systems', 'Compu
ter Vision', 'Data Science', 'Education', 'Human-Computer Interactio
n (HCI)', 'Machine Learning', 'Natural Language Processing', 'Networ
king', 'Operating/Distributed Systems', 'Programming Systems and Ver
ification', 'Robotics', 'Theory']
```

- In plotting, use the following abbreviations:

```
['ARC', 'AI', 'BIO', 'CG', 'SEC', 'SYS', 'VIS', 'DS', 'EDU', 'HCI', 'ML
', 'NLP', 'NET', 'ODS', 'PS', 'ROB', 'TH']
```

- For each research area, we want to compute how many professors works on that area.
- If one professor works on n research fields, the contribution to one research field is 1/n.
- The colors for professor ranks (assistant, associate, full professors) may be your own choice.
- Your output should be like:



In [5]:
```
# YOUR CODE MUST BE HERE
#.replace('<br/>','\\')
def cleaning(L):
    faculty = '<td class="views-field views-field-field-faculty-tit
le fac_prof_view_title">'
    focus = '<td class="views-field views-field-field-research-focu
s fac_prof_view_focus">'
    L=str(L)
    L=L.replace('<td>','').replace('</td>','')

    if faculty in L:
        L=L.replace(faculty,'')
    elif focus in L:
        L=L.replace(focus,'').replace('<br/>','&')

    L=L.replace('\n','').replace('[','').replace(']','')
    return L.split(',')

field =['Architecture', 'Artificial Intelligence', 'Computational B
iology', 'Computer Graphics', 'Computer Security', 'Computer System
s', 'Computer Vision', 'Data Science',
        'Education', 'Human-Computer Interaction (HCI)', 'Machine L
earning', 'Natural Language Processing', 'Networking', 'Operating/D
istributed Systems',
        'Programming Systems and Verification', 'Robotics', 'Theory
']

Prof, Area = [], []
for i in cleaning(soup.body.findAll('td', {"class":"views-field vie
ws-field-field-faculty-title fac_prof_view_title"})):
    Prof.append(i.strip())
for i in cleaning(soup.body.findAll('td', {"class":"views-field vie
ws-field-field-research-focus fac_prof_view_focus"})):
    Area.append(i.strip())
Dic=list(zip(Prof,Area))
print(Dic)
```

```
[('Professor', 'Computer Graphics&Human-Computer Interaction (HCI)
'), ('Professor', 'Computer Systems&Programming Systems and Verifi
cation'), ('Assistant Professor', 'Computer Systems&Operating/Dist
ributed Systems&Data Science'), ('Associate Professor', 'Computer
Systems&Architecture&Programming Systems and Verification'), ('Ass
ociate Professor', 'Artificial Intelligence&Computational Biology'
), ('Associate Professor', 'Artificial Intelligence&Computational
Biology'), ('Assistant Professor', 'Human-Computer Interaction (HC
I)'), ('Assistant Professor', 'Artificial Intelligence&Robotics'),
('Professor', 'Computer Security&Theory'), ('Assistant Professor',
'Artificial Intelligence&Machine Learning'), ('Professor', 'Theory
'), ('Professor', 'Computer Systems&Architecture'), ('Professor',
'Computational Biology&Computer Systems'), ('Associate Professor',
'Artificial Intelligence&Computational Biology'), ('Assistant Prof
essor', 'Computer Systems&Computer Security'), ('Associate Profess
or', 'Computer Systems&Programming Systems and Verification'), ('A
ssistant Professor', 'Artificial Intelligence&Machine Learning'),
('Assistant Professor', 'Computer Graphics'), ('Professor', 'Compu
ter Graphics'), ('Professor', 'Computer Systems&Data Science'), ('
Associate Professor', 'Artificial Intelligence'), ('Professor', 'A
rtificial Intelligence&Computer Graphics'), ('Professor', 'Compute
r Graphics&Computer Systems'), ('Professor', 'Computer Systems'),
('Professor', 'Computer Systems&Architecture'), ('Professor', 'Com
puter Graphics'), ('Professor', 'Artificial Intelligence&Natural L
anguage Processing'), ('Assistant Professor', 'Computer Systems&Ne
tworking'), ('Professor', 'Artificial Intelligence&Robotics'), ('A
ssociate Professor', 'Computer Systems&Architecture&Operating/Dist
ributed Systems'), ('Assistant Professor', 'Computational Biology'
), ('Professor', 'Computer Systems'), ('Professor', 'Human-Compute
r Interaction (HCI)'), ('Assistant Professor', 'Machine Learning&D
ata Science'), ('Associate Professor', 'Computer Systems&Networkin
g'), ('Associate Professor', 'Artificial Intelligence&Computer Vis
ion&Machine Learning'), ('Assistant Professor', 'Artificial Intell
igence&Machine Learning&Natural Language Processing'), ('Professor
', 'Artificial Intelligence&Natural Language Processing'), ('Profe
ssor', 'Computer Systems&Computer Security&Operating/Distributed S
ystems'), ('Professor', 'Computer Systems&Networking'), ('Professo
r', 'Computer Security'), ('Associate Professor', 'Computer System
s&Architecture&Programming Systems and Verification'), ('Associate
Professor', 'Artificial Intelligence&Machine Learning&Robotics'),
('Professor', 'Computer Systems&Architecture&Programming Systems a
nd Verification'), ('Professor', 'Computer Systems&Operating/Distr
ibuted Systems'), ('Professor', 'Computer Systems&Networking'), ('
Assistant Professor', 'Computer Systems&Data Science'), ('Professo
r', 'Theory'), ('Professor', 'Computer Systems&Operating/Distribut
ed Systems'), ('Assistant Professor', 'Theory'), ('Assistant Profe
ssor', 'Artificial Intelligence&Robotics'), ('Professor', 'Educati
on'), ('Professor', 'Artificial Intelligence&Robotics'), ('Assista
nt Professor', 'Artificial Intelligence&Computer Vision&Robotics')
, ('Professor', 'Artificial Intelligence'), ('Assistant Professor'
, 'Theory'), ('Professor', 'Artificial Intelligence&Machine Learni
ng&Robotics'), ('Assistant Professor', 'Theory'), ('Professor', 'C
omputer Systems&Data Science'), ('Assistant Professor', 'Computer
Systems&Networking'), ('Assistant Professor', 'Theory'), ('Assista
nt Professor', 'Computer Systems&Operating/Distributed Systems&Dat
a Science')]
```

Ethics:

If you cheat, you will get negatgive of the total points. If the homework total is 22 and you cheat, you get -22.

What to submit

- Run all cells
- Goto "File -> Print Preview"
- Print the page
- Submit in class
- No late homeworks accepted
- Your homework will be graded on the basis of correctness and programming skills

Deadline: 4/18