hw1- part a - most & least frequent words

In [134]:

```
fhand=open('hw1_word_counts_05.txt') #creating a handle
x=[]                          #empty list
z=[]                          #empty list
maxarray=[]                   #empty list
minarray=[]                   #empty list
probability=[]                #empty list


for lines in fhand:
    lines=lines.rstrip()      #removing linespace
    x.append(int(lines[6:]))  #list of frequencies [as from 6th index of each line the frequency is displaye
d]
    count.append(int(lines[6:]))
    z.append(lines[0:5])      #list of words
    words.append(lines[0:5])  #list of words


wordsnum=len(z)               #length of word list=6353

b=sum(x)                      #to compute prior probability {freq/sum}
for numbers in x:
    prob=numbers/b
    probability.append(prob)



i=1                            #counter
while i<16:                    #i=16 for 15 iterations

    maxfreq=max(x)             #maximum frequency in the given list
    maxind=x.index(max(x))     #index of maximum frequency

    maxarray.append(z[maxind]) #appending the max occuring words to form an new list

    x.pop(maxind)              #pop the maximum frequency from the original frequency array
    z.pop(maxind)              #pop the respective element from the original word array
    i=i+1

i=1                            #counter
while i<16:                    #i=16 for 15 iterations
```

```
    minfreq=min(x)             #minimum frequency in the given list
    minind=x.index(min(x))     #index of minimum frequency in the given list

    minarray.append(z[minind]) # appending minimum occuring words to form a new list

    x.pop(minind)              #pop the minimum frequency from the original frequency array
    z.pop(minind)              #pop the respective word element from the original word array
    i=i+1

m=1
n=1
minarray.pop()


print('Most occuring words are')        #print the most and least frequent words
for words1 in maxarray:
    print(m,words1)
    m=m+1
print('\n','least occuring words are')
for words2 in minarray:
    print(n,words2)
    n=n+1
```

```
Most occuring words are
1 THREE
2 SEVEN
3 EIGHT
4 WOULD
5 ABOUT
6 THEIR
7 WHICH
8 AFTER
9 FIRST
10 FIFTY
11 OTHER
12 FORTY
13 YEARS
14 THERE
15 SIXTY

 least occuring words are
1 BOSAK
2 CAIXA
3 MAPCO
4 OTTIS
5 TROUP
6 CCAIR
7 CLEFT
8 FABRI
9 FOAMY
10 NIAID
11 PAXON
12 SERNA
13 TOCOR
14 YALOM
```

part b - to find the most probable next letter

In [135]:

```
def egw(right, wrong, words,count):
    countz=0

    egwprob=[]                                   #empty list
    pE=[]                                        #empty list
    WORD=[]                                      #empty list


    sets='ABCDEFGHIJKLMNOPQRSTUVWXYZ'  #all the alphabets as a string

    for K in range(wordsnum):
        pinitial=1;                             #setting the initial probability for comparison of each evidence given word
        needed_set= words[K]                    #words from the list
        for L in range(0,5):
            #condition- if first letter of right word is '-' and first letter of the needed set
            if (right[L] == '-' and needed_set[L] in right) or (needed_set[L]in wrong) or (right[L] != '-' and right[L] != needed_set[L]) :
                pinitial=0
                break

        egwprob.append(pinitial)                #appending the individual probabilities into one list for the given word
        if egwprob[K] == 1:
            countz=countz + count[K]            #counting total words that have probability as 1

    i=0
    plw=[0]*26
    for i in range(wordsnum):
                if egwprob[i] != 0:             #taking only terms having value 1 as terms with 0's will be equal to 0 in product.
                    pwe = float(count[i]) / countz  #individual frequency/ total frequency= p[w|e]

                    for j in range(0,26):
        #if letter is not there in the right and not there in the wrong set but is there in the word then its probability is calculated as below
                        if not (sets[j] in right) and not (sets[j] in wrong) and (sets[j] in words[i]):
                                plw[j] += pwe

    probmax = max(plw)                      #finding maximum probability
    letter = chr(65+plw.index(probmax))     #converting it into letter after finding the letter in the maximum probability index position
    probmax=round(probmax,4)
    print('LETTER AND PROBABILITY ')
    print (letter,',', probmax)                           #printing the most probable letter along with frequency

    return


egw('-----', [],words, count)                             #calling the function for the given test cases
egw('-----', ['A','I'],words, count)
egw('A---R', [],words, count)
egw('A---R', ['E'], words, count )
egw( '--U--', ['O','D','L','C'],words, count)
egw( '-----', ['E','O'], words, count)
egw('D--I-', [],words, count)
egw('D--I-', ['A'],words, count)
egw('-U---', ['A','E','I','O','S'],words, count)
```

```
LETTER AND PROBABILITY
E , 0.5394
LETTER AND PROBABILITY
E , 0.6214
LETTER AND PROBABILITY
T , 0.9816
LETTER AND PROBABILITY
O , 0.9913
LETTER AND PROBABILITY
T , 0.7045
LETTER AND PROBABILITY
I , 0.6366
LETTER AND PROBABILITY
A , 0.8207
LETTER AND PROBABILITY
E , 0.7521
LETTER AND PROBABILITY
Y , 0.627
```