CS2121/9643 – Assignment 1
due Feb. 8, 2016 (latest to submit: Feb. 11)

1. (30pt) Order all of the following functions increasingly by growth rate (functions with the same growth rate may be ordered arbitrarily):

$$\begin{array}{ccc} 4n\log n + 2n & 2^{10} & 2^{\log n} \\ 3n + 100\log n & 4n & 2^n \\ n^2 + 10n & n^3 & n\log n \end{array}$$

2. (30pt) For the code segment below, determine the $\mathcal{O}(\cdot)$ complexity:

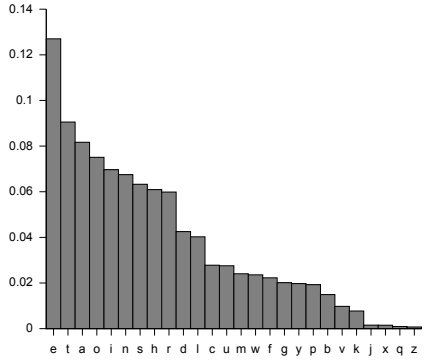
```
sum = 0
i = n
while (i >= 1) :
    sum += i
    i /= 2
i = n
j = 2
while (i >= 1) :
    sum += i
    i /= j
    j *= 2
```

(*Hint:* You do not need to find the precise complexity for all parts of the code, only for the dominant ones.)

3. (40pt) The top 10 most frequent letters in English are, in decreasing order:

e, t, a, o, i, n, s, h, r, d.

Their frequency is shown in the figure below:



Similarly, the top 10 most frequency digrams in English are, in decreasing order[1]:

th, he, in, er, an, re, ed, on, es, st

[1]Different sources give slightly different top ten letters or digrams.

Write a Python program to test this information. Your program will accept an input file (English text) and find the top 10 most frequent letters and top 10 most frequent digrams. The program will print the 10 letters *and their frequencies* in decreasing order; the same for the digrams. Note that letter case is irrelevant, that means for, example, 'th', 'Th', 'TH', and 'tH' (if any) all increment the same counter. The same for letters.

As a test file use "The 1995 CIA World Fact Book" (file "world95.txt"). In addition, you may also test it on your own input file and provide those statistics as well.

The structure of the code will not be considered for grading but it is strongly advised to implement clear logic, use meaningful names, and provide useful comments.

4. (bonus: 20pt) Give an algorithm for finding both the minimum and maximum of n numbers using at most $3n/2$ comparisons.

   (*Hint:* When comparing two elements a and b, if $a < b$, then a cannot be maximum and b cannot be minimum. Use this fact to construct a group of candidates for minimum and a group of candidates for maximum.)

Note Submit your solution on `owl.uwo.ca`:

- `lastName.py` file for question 3 and
- `lastName.pdf` file with all the answers (including the output of your program for q3).