Paragraph Analysis 2

text compression (Huffman)

Requirements

Create file in python with a **comment** containing the academic honesty pledge as shown below. Add another, separate comment to the file containing your name

- Write a python program that '*compresses*' a paragraph supplied in a textarea.
- Your code can either generate both the form (with a textarea) and the output or you can use a separate HTML file.
- Attach your python source in using the dropbox link in cobra learning.
- **Put a link to your running file in the message area**

```
# I honor Parkland's core values by affirming that I have
# followed all academic integrity guidelines for this work.

# your name
```

Compression requirements: data structures

- You do not need to split the entire text area into list of strings, you need to count the number of **each character** used in the text. Capitals, punctuation and spaces are all different characters. Use a **map** to count the occurrences of each character..
- You will create **another map** for the compression codes. The key will be the character, the value will a string representation of the binary code that character will be compressed to (like "1101").
- You will need a priority queue with frequency counts as the key and a string (or character) as the value. This will represent the combined frequency of all the characters in the string.

Huffman coding algorithm

- Place each of the frequencies of single characters into a priority_queue, low frequencies have high priority.
- while there are **more than 1 element** in the priority queue:
    - `freq1, str1 = dequeue()`
    - `freq2, str2 = dequeue()`
    - For each character in str1, prepend a 0 to its **codemap.** **`str = '0' + str`** will prepend a '0' to str.
    - For each character in str2, prepend a 1 to its **codemap**
    - `enqueue( [freq1 + freq2, str1 + str2] )`
- when there's one element on the queue it will be the full frequency and all characters.
- print out the both the frequency map and the code map in a table.

Turn in

Your python code.

A link to the website.

Links & resources

The slides on huffman coding:
https://docs.google.com/presentation/d/1-hoMkv60PG_kU6Y3NSiyd-xLZ5Mrw7EfGCKmIDh2zJo/edit?usp=sharing

http://www.csit.parkland.edu/~kurban/permanent/labs/huffman/runme.cgi is a sample site to test against your solution. **Solutions aren't unique!** You don't need to do everything that site does just what's required in the assignment.

maps needed

The frequency map is from characters to integers. **`freq['x']`** is the number of times 'x' is in the text.

The code map is from characters to strings. **`codemap['x']`** is the binary code (as a string, for printing) to replace 'x' with in compression and to change to 'x' in decompression.

```
freq = {}
codemap = {}
```