CISC106 Honors - Fall 2017
Lab 7, due Sunday

- Do labs for Understanding! You are paying for this education in some way(s), make the best of it! Use your prof and TAs to your advantage!
- Go to TA office hours!!! They are valuable. They know things. They have been there and done that.
- When posting new graphs on your website, please place the newest graphs near the **top**.
- Play with dictionaries. https://docs.python.org/3/tutorial/datastructures.html#dictionaries

Problems for submission:

1. Code merge. Code it twice: recursively (**mergeR**) and then using a while loop ((**mergeL**)[1]. Write small tests considering boundary conditions. Then write a test that tries each on two lists of size 2000. Document results.
2. Use timeit as previously to compare runs of the two merge functions. To generate sorted lists for merging, use mergesort to sort lists of sampled numbers, but be sure that happens before you start timing. Graph results to pdf of timing lists of size 100 to 500. Make lists to sort using the following:

   ```
   >>> import random
   >>> random.sample(range(1000),100)
   [163, 137, 892, 494, 37, ...
   ```

3. Code **mergesort** recursively, as done in class. Write tests.
4. Compare runtimes of your fastest mergesort and insertion sort functions on unsorted data. Graph results (to pdf) of timing lists of size 100 to 500. On the same graph, show performance of both functions on *sorted* lists of the same size. (That is four data sets on one graph.) Post to web and submit.
5. Code binary search recursively, in the form **binsearch(key, alist, start, end)**. Write good tests.
6. Time binary search looking for a random element in lists of size 10, 100, 1000, 10000, 100000, 1000000. Plot and post on your website.
7. For this and the following problem, see the file lab07.py. You may use any of the string methods described at:

   https://docs.python.org/3/library/stdtypes.html#string-methods

   Write remove_dashes. Add three tests to those provided.
8. Write remove_punctuation. Add three tests to those provided.
9. Create a dictionary. Associate animals (key) with the noise they make. Print the dictionary.

[1] Why not a for? Why not a nested for? What does a nested for usually imply about O()?