CS 161
Autumn 2017

Problem Set 0
Due: This is a fake problem set.

Exercises

Exercises should be completed **on your own.**

---

1. See the IPython notebook `HW0.ipynb` for Exercise 1. This file includes the function `estimateMean`, which we have repeated below.

```
def estimateMean(A):
    samples = []
    for i in range(10):
        samples.append(A[choice(range(len(A)))]) # draw a random sample from A
    return sum(samples)/len(samples) # return the sample mean.
```

   (a) `estimateMean(A)` attempts to estimate the mean of A. Show that the expected value that `estimateMean(A)` returns is indeed the mean of A.
   [We are expecting: A formal proof.]

   (b) In the notebook, there is some code for trying out `estimateMean(A)` a bunch of times for lists with elements between 0 and 100, and which plots the error. Based on playing around with this code, is it likely that the estimate returned by `estimateMean(A)` is off by more than 20? How likely or unlikely is this? Does your answer depend on n?
   [We are expecting: Your answers to these questions, along with a convincing empirical justification (a plot or a description of a computation you did and the outcome). You do not need to give a formal proof, just an empirical argument.]

Problems

You may talk with your fellow CS161-ers about the problems. However:

- Try the problems on your own *before* collaborating.
- Write up your answers yourself, in your own words. You should never share your typed-up solutions with your collaborators.
- If you collaborated, list the names of the students you collaborated with at the beginning of each problem.

---

1. **(Peak finding)** Given a zero-indexed array A of n integers, we say that location $i \in \{1, \ldots, n-2\}$ is a *peak* if $A[i-1] \leq A[i]$ and $A[i] \geq A[i+1]$. We say that 0 is a peak if $A[0] \geq A[1]$, and $n-1$ is a peak if $A[n-1] \geq A[n-2]$. For example, if $A = [4, 3, 5, 2, 1]$, then there are two peaks, at 0 and at 2.

   (a) Design a simple $O(n)$-time algorithm to find a peak in an array A. Notice that it does not need to return all peaks, just a single peak. In the example above with $A = [4, 3, 5, 2, 1]$, your algorithm could return either 0 or 2.
   [We are expecting: Pseudocode and a brief English description.]

   (b) Design a divide-and-conquer algorithm which finds a peak in A in time $O(\log(n))$.
   [We are expecting: Pseudocode and brief English description, as well as an informal justification of the running time. You do not need to prove that your algorithm is correct.]