ECE 20875: Python for Data Science
Fall 2020 Exam #1

Chris Brinton and Qiang Qiu

You have 100 minutes to complete and upload this 11-question exam. Assuming a worst case of 20 minutes to print and upload, that gives about 80 minutes to complete the exam. The first question is the honor pledge, so this gives you roughly **8 minutes per question**. Be sure to pace yourself accordingly. The exam is worth 100 points total, and each question is worth 10 points.

You can either print out this exam and write your answers on it, or you can write your answers on blank sheets of paper. **Do not put more than one question on the same page**. When you are finished, you will use the Gradescope app or a scanner to upload your answers. **To be safe, please allow at least 10 minutes for uploading**.

Also keep in mind the following rules:

1. For each answer, **you must show all of your work** to receive credit.
2. Under **NO** circumstances are you permitted to communicate with **ANYONE** in ECE 20875 (including the teaching staff) over the 24-hour period that the midterm is open.
3. You are **NOT** allowed to use any programming tools (e.g., calculator) or run any Python/bash scripts (e.g., Python, Jupyter notebook, PyCharm, bash, etc.) while taking the test.
4. The exam is open notes so you may use the materials on the course webpage http://www.cbrinton.net/ECE20875-2020-Fall.html or your own notes.
5. Except for the course webpage, you are **NOT** permitted to use any other online material (e.g., stackoverflow, Google search).

Question 1: Honor Pledge and Acknowledgment

Please sign the honor pledge below by uploading a signature your full legal name.

I understand and acknowledge the above instructions and notes.

I also affirm that the answers given on this test are mine and mine alone. I did not receive help from any person or material (other than those explicitly allowed).

X_____

Question 2: List Comprehensions

Suppose you have a list of numbers called `data`. Write a list comprehension that produces a new list called `new_data` that has the cubes of all the even numbers in `data` (and filters out the odd numbers).

For example, applying this list comprehension to `data = [2, -3, 5, -1, 0]` should result in `new_data = [8, 0]`.

```
new_data =
```

Question 3: Dictionaries

Complete the code in the Python `dict_swap` function below to iterate over a dictionary called `data` and swap the `values` and `keys` in the input dictionary, returning `new_data`. Since the dictionary may contain multiple `keys` with the same `value`, your code should make a `list` of such `keys` and store that `list` against the common `value` in the original dictionary for full credit.

For example, apply this program to:

```
data = {'director': 'Nolan', 'language': 'English', 'movie': 'Tenet', 'writer': 'Nolan'}
```

should result in:

```
new_data = {'English': ['language'], 'Nolan': ['director', 'writer'], 'Tenet': ['movie']}
```

```
def dict_swap(data):

    new_data = dict()









    return new_data
```

Question 4: Data Structures

Fill in the missing code for the `moving_list` function below. This function should take the input list `data` of numbers of length `N` and return two output lists:

The first output, `pack_data`, should be a list of tuples, where tuple `i` is defined as follows:

- The first tuple (`i = 0`) should contain the first two entries in `data`.
- The last tuple (`i = N-1`) should contain the last two entries in `data`.
- The middle tuples (`i = 1,…,N-2`) should contain the `i-1`th, `i`th, and `i+1`th entries from `data`.

The second output, `sum_pack`, should be a list of the sum of the numbers in the tuples. (This is one way of computing moving averages in statistics.)

For example, running `moving_list([1, 2, 3, 4, 5])` should return `pack_data = [(1,2), (1,2,3), (2,3,4), (3,4,5), (4,5)]` and `sum_pack = [3, 6, 9, 12, 9]`.

```
def moving_list(data):









    return
```

Question 5: Map/Reduce

Suppose we have a list of strings and numbers called `str_num_list`. We want to concatenate the strings and the numbers in the list into a single string `str_conc` where the original strings are separated by a space. For example, if `str_num_list = ['abc', 2, 34, 'def']`, then `str_conc = 'abc 2 34 def'`.

Complete the single line of code within the Python `reduce()` function below to implement this operation, using the `lambda` function.

```
from functools import reduce

str_conc = reduce(                                    )
```

Question 6: Function Compositions

Suppose you are the grader for a homework assignment. You have two dictionaries, `name_to_score` and `name_to_latedays`. `name_to_score` contains key-value pairs of `{student name: student score}`, and `name_to_latedays` contains the key-value pairs `{student name: number of days late}`. For each day late a student submitted their assignment, their score is adjusted as specified in a helper function `fun()`.

Fill in the missing code in the `adjust_score(name_to_score, name_to_latedays)` function below to determine grades. This function takes in the dictionaries above and returns a new dictionary `adj_scores` with student's scores adjusted for any penalties incurred for late submission. **You must use the concept of function composition to solve this problem**, using repeated calls to `fun()`.

For example, applying this program to:

```
name_to_score = {'Daniel Jackson': 100, 'Samantha Carter':98, 'Tealc': 60}
name_to_latedays = {'Daniel Jackson':0, 'Samantha Carter':1, 'Tealc': 5}

fun(score):
{
     adj_score = score – 10
     return adj_score
}
```

Should output `{'Daniel Jackson':100, 'Samantha Carter':88, 'Tealc': 10}`.

```
def adjust_score(name_to_score, name_to_latedays, fun):
     adj_scores = {}
     for name in name_to_score:
          days_late = name_to_latedays[name]
          score = name_to_score[name]









          adj_scores[name] = score

     return adj_scores
```

Question 7: Recursion

The Fibonacci numbers form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is,

$F_0 = 0, F_1 = 1,$

and

$F_n = F_{n-1} + F_{n-2}$, for $n>1$.

The first several numbers in the sequence is thus: 0, 1, 1, 2, 3, 5, 8, 13, …

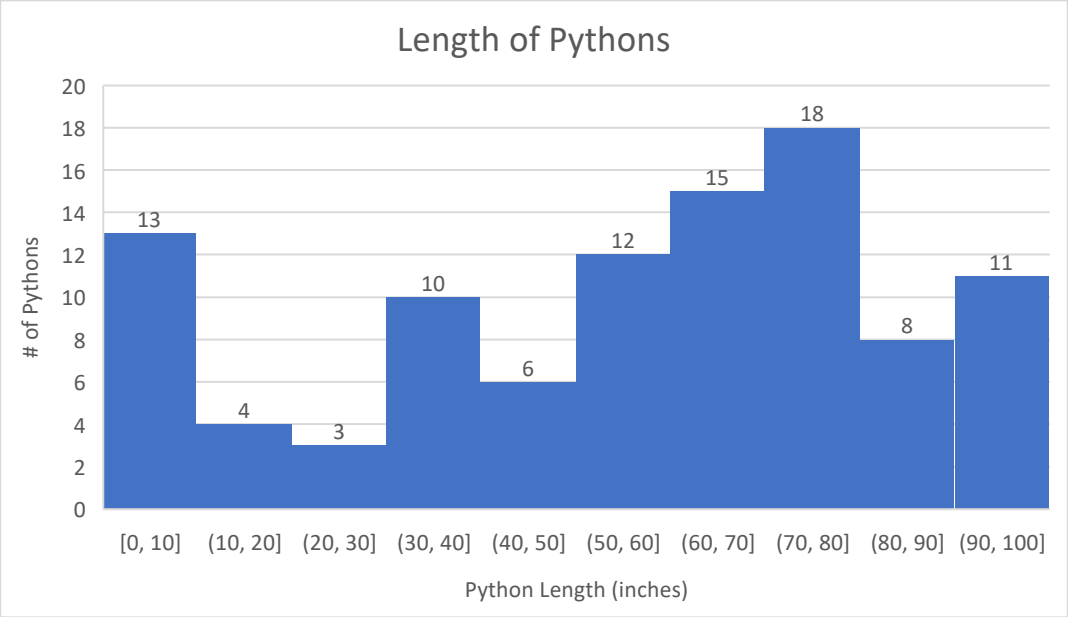
Complete the following Python function `fib(n)` to compute term `n` in the Fibonacci sequence (starting at `n = 0`). **You must use recursion as part of your answer**, i.e., with the `fib` function making a call to itself.

For example, calling `fib(4)` should return `Fn = 3`.

```
def fib(n):









    return Fn
```

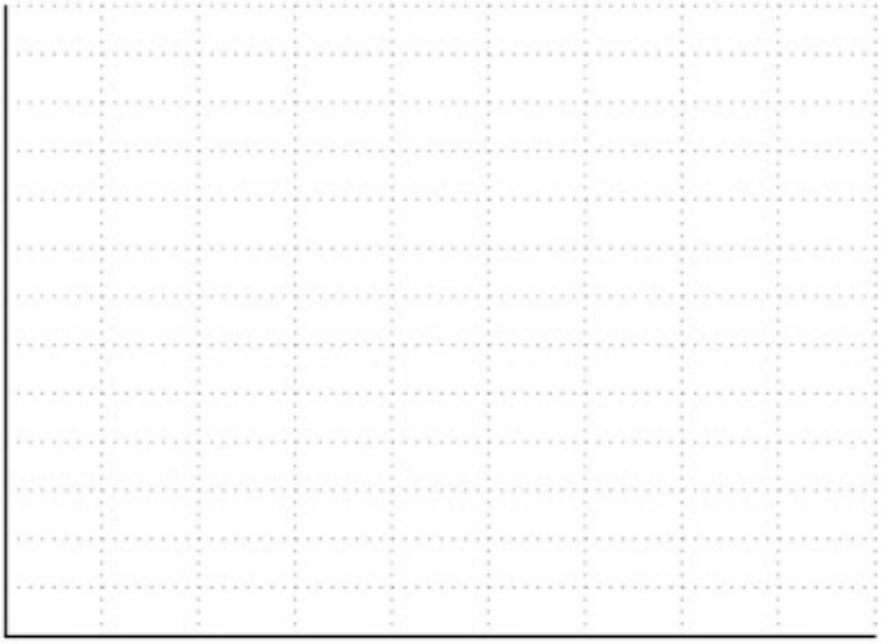
Question 8: Histograms

The Indianapolis Zoo is conducting a study of their ball pythons to see if length and morph type are related. They started by measuring the length of all 100 pythons in their reptile exhibit. The histogram below represents the distribution of python lengths in inches, using a bin width of 10 inches.



A. The Zoo wants to be able to compare their python length distribution with other zoos in the US. Normalize the histogram above using density normalization and write the normalized value for each bin below so the Zoo can compare percentages. (Note: you do not have to redraw the histogram, just write the normalized values.)

B. The Zoo is afraid that they have too many bins to draw conclusions from. Redraw the original histogram with only 5 bins on the graph below, using count on the y-axis. Be sure to keep a uniform bin width, and to label your y-axis values carefully.



C. The Zoo knows that the average (mean) length of their Pythons is 55 inches. Is it possible to know how many pythons at the Zoo are equal to or longer than this value? If so, how many? Explain your answer in 1-2 sentences.

Question 9: Sampling

We are monitoring usage of a wireless network on campus. After collecting data, we find that in any given minute, the probability that the network is occupied follows a Bernoulli random variable X with a success probability of $p = 0.8$. Recall that a Bernoulli random variable X has a mean $\mu_X = p$ and a variance $\sigma_X^2 = p(1 - p)$.

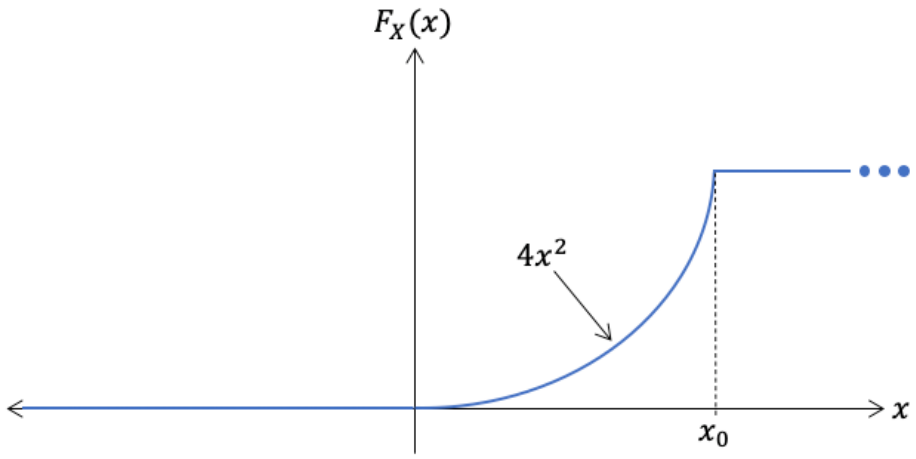
A. If we sampled 100 minutes of this network independently and at random, what is the total number of times we would expect to find that the network is *not* occupied?

B. If we sampled 100 minutes of this network independently and at random, what is a good approximation of the distribution for $\bar{X}$, the fraction of minutes we would find the network is occupied?

C. Is X a continuous or discrete random variable? How about $\bar{X}$? Explain in 1-2 sentences.

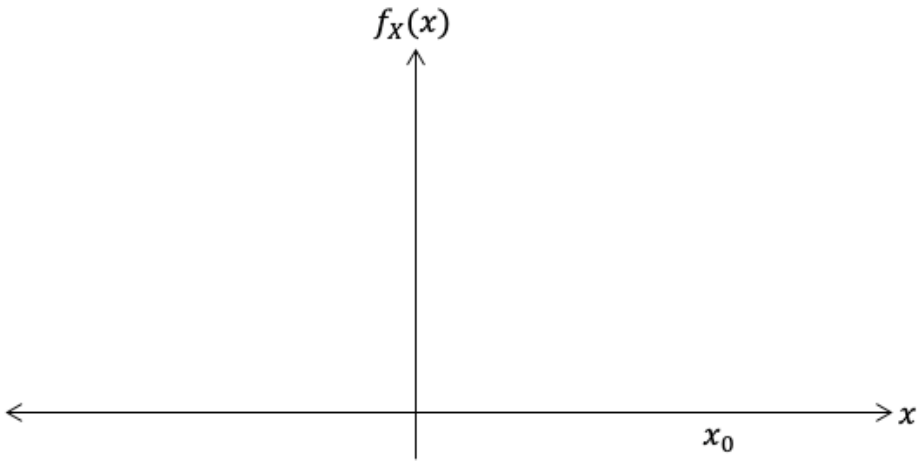
Question 10: Probability Distributions

Consider a random variable X with the CDF graphed below.



A. What must be the value of x_0?

B. Determine the PDF $f_X(x)$ of this random variable mathematically, and sketch it on the graph below. Be sure to consider all possible values of X, and to label critical values on the x and y axes.



C. The mean of a continuous random variable X can be obtained from its PDF as

$$E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$

(i.e., a weighted integral of the PDF by x.) What is the mean of the variable X in this problem?

Question 11: Hypothesis Testing

The mean weight of White Feather chickens at five months is known to be 435, measured in hundredths of pounds. We assume the weights follow the normal distribution with a known standard deviation of 16. In an effort to change the weights, a special feed is given to a sample of 64 chickens. The subsequent average weight of this sample is 439.

We want to answer the question: *Does the special feed have any effect (either positive or negative) on the mean weight of the chickens*?

A. Formulate the null and alternative hypotheses for this statistical test.

B. What is the z-score of the statistical test?

C. Find the p-value (hint: remember the 68-95-99.7% rule) of the test and sketch the result on a normal curve below. Is the result significant at $\alpha = 0.1$? How about 0.01?