Asmt 1: Hash Functions and PAC Algorithms

Turn in (**a pdf**) through Canvas by 2:45pm:
Wednesday, January 18

Overview

In this assignment, you will experiment with random variation over discrete events. It will be very helpful to use the analytical results and the experimental results to help verify the other is correct. If they do not align, you are probably doing something wrong (this is a very powerful and important thing to do whenever working with real data).

Note: Homework assignments are intended to help you learn the course material, and successfully solve mid-term and final exams that will be done on paper in person.

As usual, it is recommended that you use LaTeX for this assignment. If you do not, you may lose points if your assignment is difficult to read or hard to follow. The latex source for this homework is available on Canvas. I recommend that you drop the two latex files (.tex and .sty) in a new Overleaf project, and compile/edit the .tex file there.

- **LaTeX tip #1:** Use "lstlisting" for sharing Python code.

- **LaTeX tip #2:** Include figures with "includegraphics".

1 Birthday Paradox (35 points)

Consider a domain of size $n = 10{,}000$.

A: (5 points) Generate random numbers in the domain $[n]$ until two have the same value. How many random trials did this take? We will use k to represent this value.

Answer: k = 39 random trials is needed until two have the same value within the domain size of 10000.

B: (10 points) Repeat the experiment $m = 500$ times, and record for each time how many random trials this took. Plot this data as a *cumulative density plot* where the x-axis records the number of trials required k, and the y-axis records the fraction of experiments that succeeded (a collision) after k trials. The plot should show a curve that starts at a y value of 0, and increases as k increases, and eventually reaches a y value of 1.

Answer:

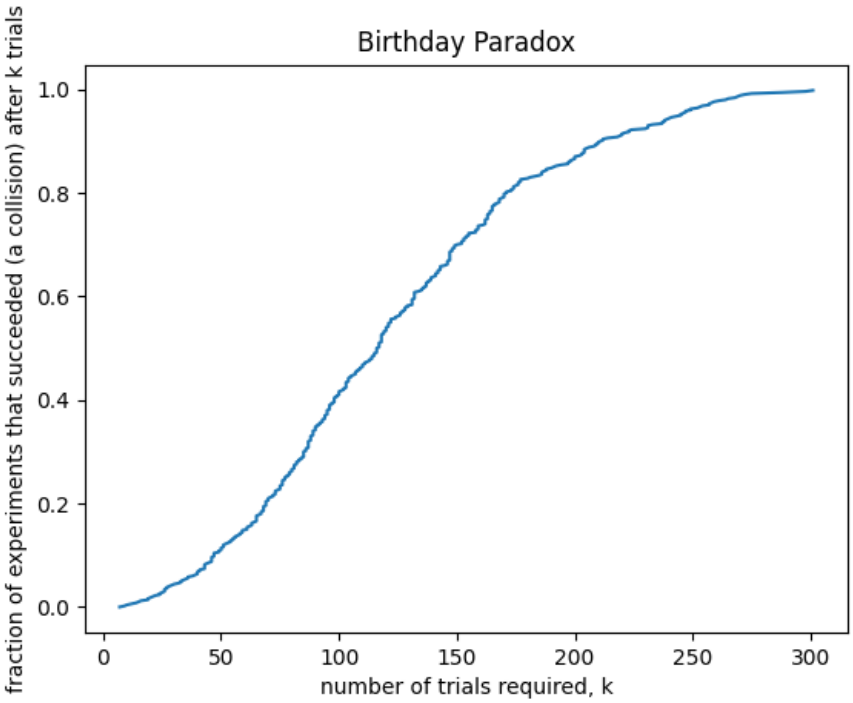

Figure 1: Graph showing the cumulative density that records the number of trials required represented by k against fraction of experiments succeeded after k trials for Birthday Paradox problem

The graph shows a curve that starts from 0 and eventually increases to 1. This is done using m = 500 and n = 10000.

C: (10 points) Empirically estimate the expected number of k random trials in order to have a collision. That is, add up all values k, and divide by m.

Answer: Expected number of k random trials in order to have a collision = 122.762

D: (10 points)

1. Show how you implemented the experiment in **1.C**.
2. How long did **1.C** take for $n = 10,000$ and $m = 500$ trials?
3. Show a plot of the run time as you gradually increase the parameters n and m. (For at least 3 fixed values of m between 500 and 10,000, plot the time as a function of n.) You should be able to reach values of $n = 1{,}000{,}000$ and $m = 10{,}000$.

Answer (i):

```
def q1bc(m,n):
    start = time.time()

    k_lst = []

    for i in range(m):
        k_lst.append(q1a(n))
```

```
    x = np.sort(k_lst)
    y = np.arange(m)/m

    end = time.time()

    avg = np.sum(k_lst)/m

    print("time ", end-start)

    return x, y, avg
```

avg will produce the expected number of k random trials in order to have a collision.
this is the output produced:

```
Birthday Paradox
Expected number of k random trials in order to have a collision: 122.762
```

Answer (ii): For m = 500 and n = 10000 for birthday paradox problem, the recorded time taken for the run time in seconds is as follows:

```
time   0.07624220848083496
```

Answer (iii):

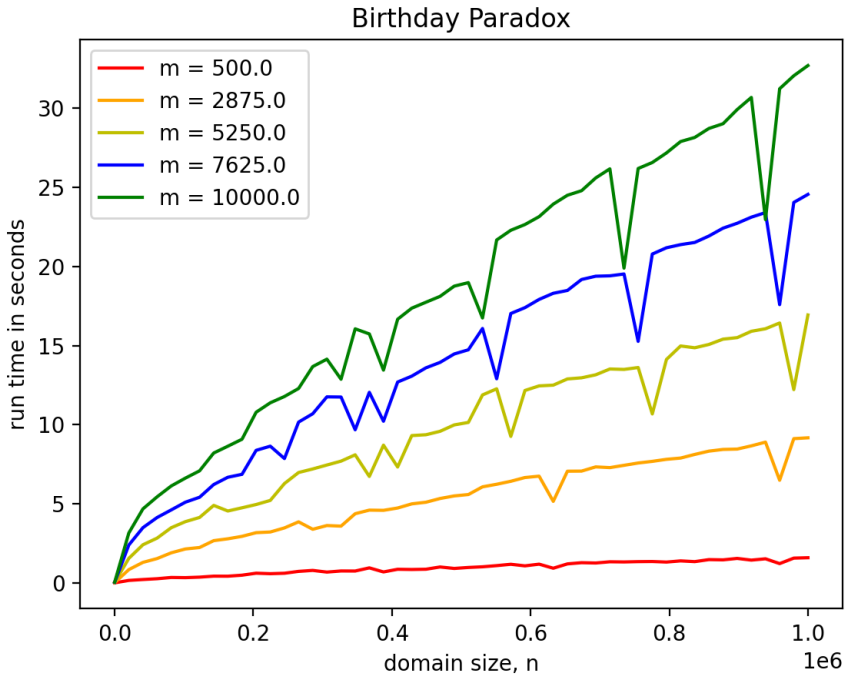

Figure 2: Graph showing the run time for a range of m values for Birthday Paradox problem

The graph in the figure above shows the run time for m = [500, 1625, 2750, 3875, 5000] with a range of n values spread over 50 values from 1 to 1e6.

The anomaly points seen in several points of n, such as in 0.6 or 0.8 is suspected due to slight disturbance in the CPU working as the program runs. This might be because the computer is also used to browse through the internet at the time.

2 Coupon Collectors (35 points)

Consider a domain $[n]$ of size $n = 1{,}000$.

A: (5 points) Generate random numbers in the domain $[n]$ until every value $i \in [n]$ has had one random number equal to i. How many random trials did this take? We will use k to represent this value.

Answer: k = 6993 random trials needed for every element within that domain of size 1000 has had one random number equal to it.

B: (10 points) Repeat step **A** for $m = 500$ times, and for each repetition record the value k of how many random trials we required to collect all values $i \in [n]$. Make a cumulative density plot as in **1.B**.

Answer:

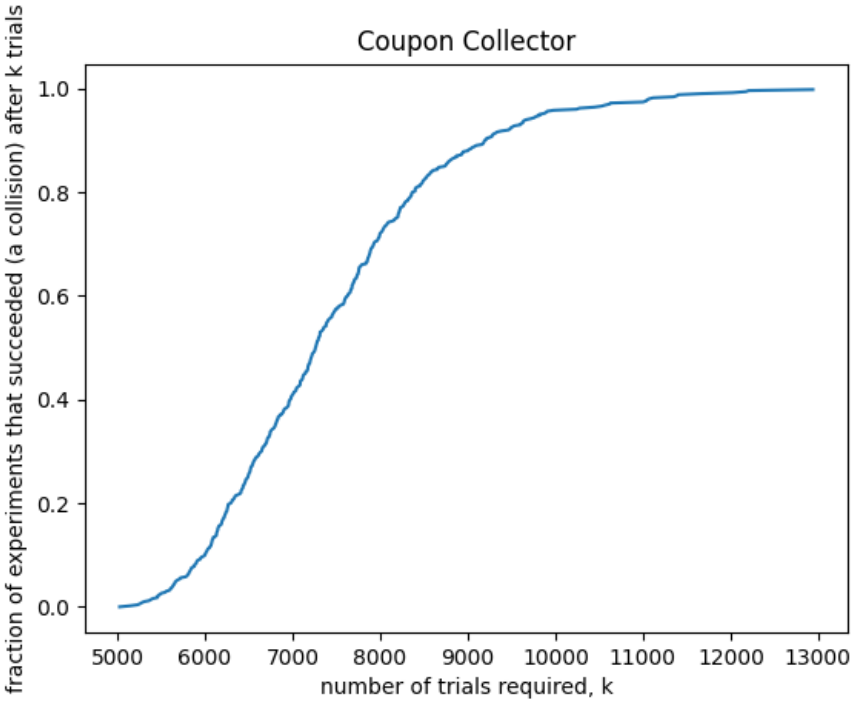

Figure 3: Graph showing the cumulative density that records the number of trials required represented by k against fraction of experiments succeeded after k trials for Coupon Collector problem

C: (10 points) Use the above results to calculate the empirical expected value of k.

Answer: Expected number of k random trials in order to have a collision for every value in the domain = 7446.006

D: (10 points)

1. Show how you implemented the experiment in **2.C**.

2. How long did **2.C** take for $n = 1,000$ and $m = 500$ trials?

3. Show a plot of the run time as you gradually increase the parameters n and m. (For at least 3 fixed values of m between 500 and 5,000, plot the time as a function of n.) You should be able to reach $n = 20{,}000$ and $m = 5{,}000$.

Answer (i):

```
def q2bc(m,n):
    start = time.time()
    k_lst = []

    for i in range(m):
        k_lst.append(q2a(n))

    x = np.sort(k_lst)
    y = np.arange(m)/m

    end = time.time()

    avg = np.sum(k_lst)/m

    print("time", end-start)
    return x, y, avg
```

Similar to birthday paradox, **avg** will produce the expected number of k random trials for every elements in the domain to have a collision. The output produce is such follows:

```
Coupon Collector
Expected number of k random trials in order to have a collision: 7446.006
```

Answer (ii): For m = 500 and n = 1000 for coupon collector paradox problem, the recorded time taken for the run time in seconds is as follows:

```
time 3.9632246494293213
```

Answer (iii):

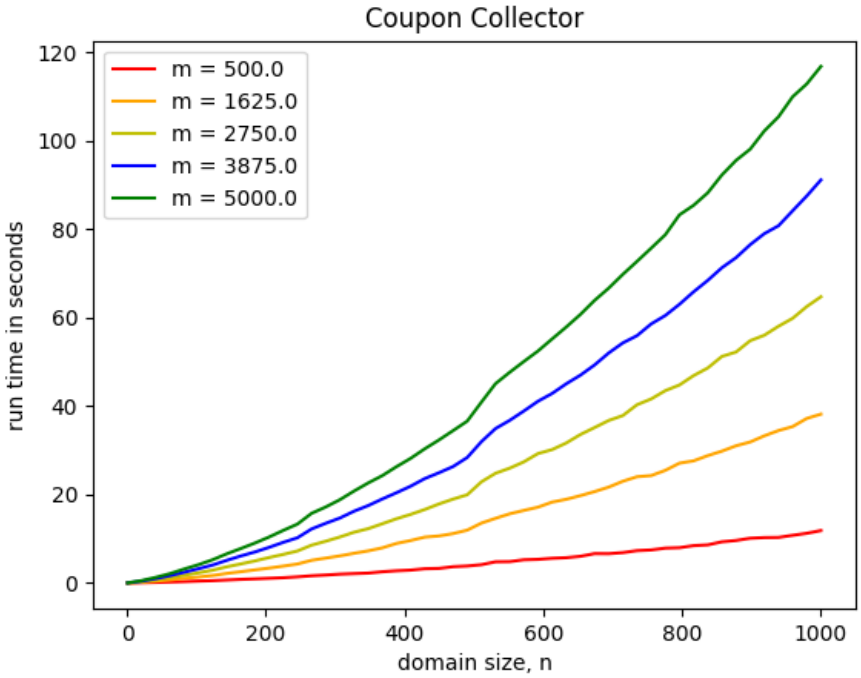

Figure 4: Graph showing the run time for a range of m values for Coupon Collector problem

The graph in the figure above shows the run time for m = [500, 1625, 2750, 3875, 5000] with a range of n values spread over 50 values from 1 to 1000.

This maximum n value has been reduced from previous testing case because the program was taking too long for previously higher n values, such as: 10000 or 11000.

3 Comparing Experiments to Analysis (30 points)

A: (15 points) Calculate analytically (using the formula from the lecture) the number of random trials needed so there is a collision with probability at least 0.5 when the domain size is $n = 10{,}000$. How does this compare to your results from **1.C**?

[Show your work, including describing which formula you used.]

Answer:

As discussed in the lecture, the probability that there is no collision occurs with k pairs is described as:

$$\mathbb{P}(\text{no collision with k random trials}) = \left(1 - \frac{1}{n}\right)^{\binom{k}{2}(1)}$$

Thus, to calculate the number of random trials needed to produce a collusion can be defined as:

$$\begin{aligned}\mathbb{P}(\text{collision with random trials}) &= 1 - \mathbb{P}(\text{no collision with k random trials}) \\ &= 1 - \left(1 - \frac{1}{n}\right)^{\binom{k}{2}}\end{aligned}$$

In this case, calculating the number of random trials needed so there is a collision with probability at least 0.5 when the domain size is $n = 10{,}000$ can be solved by:

$$\mathbb{P}(\text{collision with k random trials}) = 1 - \left(1 - \frac{1}{n}\right)^{\binom{k}{2}}$$

$$1 - \left(1 - \frac{1}{n}\right)^{\binom{k}{2}} \geq 0.5$$

$$1 - \left(1 - \frac{1}{10000}\right)^{\binom{k}{2}} \geq 0.5$$

$$\left(1 - \frac{1}{10000}\right)^{\binom{k}{2}} \leq 0.5$$

$$\binom{k}{2} = \log_{(1-\frac{1}{10000})} 0.5$$

$$\binom{k}{2} = \log_{\frac{9999}{10000}} 0.5 = 6931$$

So if the approximation of $\binom{k}{2}$ is 6931 with trial and error, k can be deduced to either 118 or 119. Then, inputting those 2 possible values into the inequalities above(k = 118: $\mathbb{P}$(collision with k random trials) = 0.498, k = 119: $\mathbb{P}$(collision with k random trials) = 0.504), only 119 that produce the probability above 0.5. So, k is 119.

B: (15 points) Calculate analytically (using the formula from the lecture) the expected number of random trials before all elements are witnessed in a domain of size $n = 1,000$? How does this compare to your results from **2.C**?
[Show your work, including describing which formula you used.]

Answer:

From the lecture, it is known that the expected number of trials to get all the coupons is:

$$T = \sum_{i=1}^{n} t_i = \sum_{i=1}^{n} \frac{n}{n-i+1} = n \sum_{i=1}^{n} \frac{1}{i} \tag{2}$$

With domain size of 1000, then calculated expected number of trials to get all the coupons can be described as:

$$T = n \sum_{i=1}^{n} \frac{1}{i}$$
$$= 1000 \sum_{i=1}^{1000} \frac{1}{i}$$

The term $\sum_{i=1}^{n} \frac{1}{i}$ is known at the n-th Harmonic Number, H_n, which is equivalent to $\gamma + ln(n)$ in which γ, Euler-Masheroni constant, is 0.577.

So, inputting this into the equation above:

$$T = n \sum_{i=1}^{n} \frac{1}{i}$$
$$= 1000 \sum_{i=1}^{1000} \frac{1}{i}$$
$$= nH_n \qquad = n(\gamma + ln(n)) = 1000(0.577 + ln(1000)) = 7484.755$$

This corresponds positively expected number of k random trials in order to have a collision in Coupon Collector problem for n = 1000 and m = 500 experiments = 7449.318, which has a insignificant difference with the calculated expected number.

4 BONUS : PAC Bounds (2 points)

Consider a domain size n and let k be the number of random trials run, where each trial obtains each value $i \in [n]$ with probability $1/n$. Let f_i denote the number of trials that have value i. Note that for each $i \in [n]$ we have $\mathbf{E}[f_i] = k/n$. Let $\mu = \max_{i \in [n]} f_i/k$.

Consider some parameter $\varepsilon \in (0, 1)$. As a function of parameter ε, how large does k need to be for $\mathbf{Pr}[|\mu - 1/n| \geq \varepsilon] \leq 0.05$? That is, how large does k need to be for *all* counts to be within $(\varepsilon \cdot 100)\%$ of the average with probability 0.05? *(Fine print: you don't need to calculate this exactly, but describe a bound as a function of ε for the value k which satisfies PAC property. Chapter 2.3 in the M4D book should help.)*

How does this change if we want $\mathbf{Pr}[|\mu - 1/n| \geq \varepsilon] \leq 0.005$ (that is, only 0.005 probability of exceeding ε error)?

[Make sure to show your work.]

Answer: