1. *An urn contains 4 white balls and 4 black balls. 2 balls are drawn randomly (without replacement) from the urn. If they are the same color, your win \$2. If they are different colors, you lose \$1 (i.e., you win -\$1). Let X equal the amount you win.*

   (a) *What is $E[X]$?*

$$P(\text{2 balls are the same color}) = \frac{3}{7} = .429$$
$$P(\text{2 balls are different colors}) = \frac{4}{7} = .571$$
$$E[X] = \sum_{x:P(x)>0} xP(x) = \$2 * \frac{3}{7} + (-\$1) * \frac{4}{7} = \$0.29$$

   (b) *What is $Var[X]$?*

$$Var[X] = E[X^2] - E[X]^2 = (\$2)^2 * \frac{3}{7} + (-\$1)^2 * \frac{4}{7} - E[X]^2 = \$2.20$$

2. *Say there are k buckets in a hash table. Each new string added to the table is hashed to bucket i with probability p_i, where $\sum_{i=1}^{k} p_i = 1$. If n strings are hashed into the table, find the expected number of buckets that have at least one string hashed to them. (Hint: Let X_i be a binary variable that has the value 1 when there is at least one string hashed to bucket i after the n strings are added to the table (and 0 otherwise). Compute $E[\sum_{i=1}^{k} X_i]$.)*

   A_i is the event that at least one string is hashed to bucket i

$$X_i = \mathbb{1}[A_i] \begin{cases} 1 & \text{if } A_i \text{ occurs} \\ 0 & \text{otherwise} \end{cases}$$

$$P(X_i) = 1 - (1 - p_i)^n$$
$$E[X_i] = 1 * P(X_i) = 1 - (1 - p_i)^n$$
$$E[\sum_{i=1}^{k} X_i] = k * E[X_i] = k * (1 - (1 - p)^n)$$

3. *Recall the coin-flipping game set-up discussed in class (called the "St. Petersburg Paradox"): There is a fair coin which comes up "heads" with a probability $p = 0.5$. The coin is flipped repeatedly until the first "tails" appears. Let N be the number of coin flips before the first "tails" appears (i.e. N is the number of consecutive "heads" that appear). Given that no one really has infinite money to offer as payoff for the game, consider a variant of the game where you win $min(\$2^N, X)$, where X is the maximum amount that the game provider will pay you after playing. Compute the expected payoff of the game for the following values of X.*

   W is the amount of money won
   H_i event flip comes up "heads"
   T_i event flip comes up "tails"

   (a) $X = \$20$

$$W = \begin{cases} \$2^N & \text{if } N \in \mathbb{Z}, N < 5 \\ \$20 & \text{if } N \in \mathbb{Z}, N \geq 5 \end{cases}$$
$$P(N \geq 5) = P(H_1H_2H_3H_4H_5) = p^5 = .03125$$
$$E[W] = \$2^0P(T_1) + \$2P(H_1T_2) + \$2^2P(H_1H_2T_3) + \$2^3P(H_1H_2H_3T_4) + \$2^4P(H_1H_2H_3H_4T_5) + \$20P(H_1H_2H_3H_4H_5)$$
$$= \sum_{j=0}^{4}(\$2^j p^{j+1}) + \$20p^5 = \$3.13$$

(b) $X = \$500$

$$W = \begin{cases} \$2^N & \text{if } N \in \mathbb{Z}, N < 9 \\ \$500 & \text{if } N \in \mathbb{Z}, N \geq 9 \end{cases}$$
$$P(N \geq 9) = p^9 = .001953$$
$$E[W] = \sum_{j=0}^{8} (\$2^j p^{j+1}) + \$500p^9 = \$5.48$$

(c) $X = \$10,000$

$$W = \begin{cases} \$2^N & \text{if } N \in \mathbb{Z}, N < 14 \\ \$500 & \text{if } N \in \mathbb{Z}, N \geq 14 \end{cases}$$
$$P(N \geq 14) = p^{14} = .001953$$
$$E[W] = \sum_{j=0}^{13} (\$2^j p^{j+1}) + \$500p^{14} = \$7.61$$

4. *Say we have an integer array "arr[10]" (indexed from 0 to 9), which contains the numbers 1 through 10 in sorted order. Now, say the key is a randomly generated integer value between 1 and 10, inclusive, where each value is equally likely.*

   (a) *What is the expected number of times that the "equality test" (as noted by the comment in the code) is executed in the function "linear" (assuming "linear" is passed the array "arr" and the randomly chosen value "key")?*

```
int linear(int arr[], int key)
{
    for (int i = 0; i < 10; i++)
    {
        // Equality test: (arr[i] == key)
        if (arr[i] == key)
        {
            return i;
        }
    }
    return -1; // Will never get here when key is in [1, 10]
}
```

   X_k is the number of times the "equality test" is run for $key = k$
   X is the number of times the "equality test" is run for a key = [1, 10]

$$X_k = k$$
$$P(\text{key} = \text{k}) = .1$$
$$E[X] = \sum_{i=1}^{10} i * .1 = 5.5$$

   (b) *Under the same conditions for array "arr" and the randomly chosen value "key", what is the expected number of times that the "equality test" is executed in the function "binary" below?*

```
int binary(int arr[], int key)
{
    int low = 0;
    int high = 9;
```

```
    while (low <= high)
    {
        int mid = (low + high) / 2;
        // Equality test: (arr[mid] == key)
        if (arr[mid] == key)
        {
            return mid;
        }
        else if (arr[mid] < key)
        {
            low = mid + 1;
        }
        else
        {
            high = mid - 1;
        }
    }
    return -1; // Will never get here when key is in [1, 10]
}
```

key	tests
1	3
2	2
3	3
4	4
5	1
6	3
7	4
8	2
9	3
10	4

$$E[X] = 1 * \frac{1}{10} + 2 * \frac{2}{10} + 3 * \frac{4}{10} + 4 * \frac{3}{10} = 2.9$$

5. *When a bit string is sent over a network, each bit in the string will independently be corrupted (flipped) with a probability p. Say we come up with a protocol for sending strings over the network where if we have an original string s of length n bits, we can create the message ss (just two copies of the original message in a row, so ss has length 2n bits) and send that message over the network instead. Thus, the recipient can detect an error if there are any discrepancies between the first and second halves of the string they receive. Note that it is possible for the recipient to not be able to detect an error if a bit and its corresponding duplicate in the second half of the message are both corrupted (flipped).*

   (a) *What is the expression (in terms of n and p) for the probability that the message ss is received without any corruption? Also, compute the numerical value for your expression for* $n = 5$ *and* $p = 0.1$.

$$\begin{aligned} P(\text{no errors}) &= (1-p)^{2n} \\ &= .349 \qquad \text{for } n = 5, p = 0.1 \end{aligned}$$

   (b) *What is the expression (in terms of n and p) for the probability that the recipient receives a corrupted message and is not able to detect that it is corrupted? Also, compute the numerical value for your expression for* $n = 5$ *and* $p = 0.1$.

X_i is event where the same i bits are flipped in the first and second n bits

$$P(\text{errors undetected}) = \sum_{i=1}^{n} \binom{n}{i} * (p^{2i}(p-1)^{2(n-i)})$$

$$P(\text{errors undetected}) = \binom{5}{1}p^2(1-p)^8 + \binom{5}{2}p^4(1-p)^6 + \binom{5}{3}p^6(1-p)^4 + \binom{5}{4}p^8(1-p)^2 + \binom{5}{5}(1-p)^{10} = .022 \quad \text{for } n = 5, p = 0.1$$

6. *Suppose it takes at least 9 votes from a 12-member jury to convict a defendant. Suppose also that the probability that a juror votes that an actually guilty person is innocent is 0.25, whereas the probability that the juror votes that an actually innocent person is guilty is 0.15. If each juror acts independently and if 70% of defendants are actually guilty, find the probability that the jury renders a correct decision. Also determine the percentage of defendants found guilty by the jury.*

$p_{wrong,g}$ is the probability a juror votes that an actually guilty person is innocent = .25
$p_{wrong,i}$ is the probability a juror votes that an actually innocent person is guilty = .15

$$P(\text{correct decision}) = P(\text{convicted}|\text{guilty})P(\text{guilty}) + P(\text{acquitted}|\text{innocent})P(\text{innocent})$$

$$P(\text{convicted}|\text{guilty}) = \sum_{i=9}^{12} \binom{12}{i}(1-p_{wrong,g})^i p_{wrong,g}^{12-i}$$
$$= \binom{12}{9}(1-p_{wrong,g})^9 p_{wrong,g}^3 + \binom{12}{10}(1-p_{wrong,g})^{10} p_{wrong,g}^2$$
$$+ \binom{12}{11}(1-p_{wrong,g})^{11} p_{wrong,g} + \binom{12}{12}(1-p_{wrong,g})^{12} = .649$$

$$P(\text{convicted}|\text{innocent}) = \sum_{i=9}^{12} \binom{12}{i} p_{wrong,i}^i (1-p_{wrong,i})^{12-i}$$
$$= \binom{12}{9} p_{wrong,i}^9 (1-p_{wrong,i})^3 + \binom{12}{10} p_{wrong,i}^{10} (1-p_{wrong,i})^2$$
$$+ \binom{12}{11} p_{wrong,i}^{11} (1-p_{wrong,i}) + \binom{12}{12} p_{wrong,i}^{12} = 5.48 \times 10^{-6}$$

$$P(\text{correct decision}) = .754$$

$$P(\text{convicted}) = P(\text{convicted}|\text{guilty})P(\text{guilty}) + P(\text{convicted}|\text{innocent})P(\text{innocent})$$
$$= .454$$

7. *Consider a hash table with n buckets. Now, m strings are hashed into the table (with equal probability of being hashed into any bucket).*

   (a) *Let $n = 2000$ and $m = 10000$. What is the (Poisson approximated) probability that the first bucket has 0 strings hashed to it?*

X is the number of strings assigned to bucket 0

$$\lambda = \text{average number of strings assigned to each bucket} = \frac{m}{n} = 5$$

$$P(X = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$

$$P(X = 0) = \frac{5^0}{0!}e^{-5} = 6.738 \times 10^{-3}$$

(b) *Let $n = 2000$ and $m = 10000$. What is the (Poisson approximated) probability that the first bucket has 8 or fewer strings hashed to it?*

$$P(X \leq 8) = \sum_{i=0}^{8} \frac{\lambda^i}{i!} e^{-\lambda} = \sum_{i=0}^{8} \frac{5^i}{i!} e^{-5} = .932$$

(c) *Let $m = 10000$. What is the largest integer value n such that the (Poisson approximated) probability that an arbitrary bucket in the hash table will have no strings hashed to it is less than 0.5?*

$$\lambda = \frac{m}{n} = \frac{10000}{n}$$
$$P(X = 0) = \frac{\lambda^0}{0!} e^{-\lambda} = e^{-\frac{10000}{n}} \leq 0.5$$
$$n = 14426$$

(d) *Let X be a Poisson random variable with parameter λ, that is: X $Poi(\lambda)$. What value of λ maximizes $P(X = 3)$? Show formally (mathematically) how you derived this result.*

$P(X = 3)$ extrema occur where $\frac{d}{d\lambda} P(X = 3) = 0$

$$P(X = 3) = \frac{\lambda^3}{3!} e^{-\lambda}$$
$$\frac{d}{d\lambda} P(X = 3) = \frac{d}{d\lambda}(\frac{\lambda^3}{6}) e^{-\lambda} + \frac{\lambda^3}{6} \frac{d}{d\lambda}(e^{-\lambda}) = -\frac{1}{6} e^{-\lambda} \lambda^2 (\lambda - 3) = 0$$

maxima occurs at $\lambda = 3$

8. *Consider a computer cluster (data center) of 100 web servers, where incoming requests are randomly assigned to servers with equal probability. Based on historical averages, each server in the data center receives requests at a rate of 2 per second. Some buggy server code was just deployed to all the servers in the cluster and as a result any server will crash if it receives more than 6 requests in a second. What is the approximate probability that no servers have crashed 1 second after the buggy code is deployed?*

$p_{c,i}$ is the probability that a single web server will crash
$p_{nc,dc}$ is the probability that no web server will crash in the data center
X_i is the number of requests a single web server receives per second

$X_i \sim Poi(\lambda)$ where $\lambda = 2$

$$p_{c,i} = P(X \geq 6) = 1 - P(X = 5) = 1 - \sum_{j=0}^{5} \frac{\lambda^j}{j!} e^{-\lambda} = 1.66 \times 10^{-2}$$

$$p_{nc,dc} = (1 - p_{c,i})^{100} = .188$$

9. *The number of times a person's computer crashes in a month is a Poisson random variable with $\lambda = 7$. Suppose that a new operating system patch is released that reduces the Poisson parameter to $\lambda = 2$ for 80% of computers, and for the other 20% of computers the patch has no effect on the rate of crashes. If a person installs the patch, and has his/her computer crash 4 times in the month thereafter, how likely is it that the patch has had an effect on the user's computer (i.e. it is one of the 80% of computers that the patch reduces crashes on)?*

X number of computer crashes in a month λ_e number of computer crashes in a month if patch effective

$= 2$ λ_n number of computer crashes in a month if patch not effective $= 7$

$$P(\text{computer patch effective}) = \frac{P(X = 4|\text{effect})P(\text{effect})}{P(X = 4|\text{effect})P(\text{effect}) + P(X = 4|\text{no effect})P(\text{no effect})}$$

$$P(X = 4|\text{effect}) = \frac{\lambda_e^4}{4!}e^{-\lambda_e} = 9.02 \times 10^{-2}$$

$$P(X = 4|\text{no effect}) = \frac{\lambda_n^4}{4!}e^{-\lambda_n} = 9.12 \times 10^{-2}$$

$$P(\text{computer patch effective}) = .798$$

10. *An election has 2 candidates in a very close race: recent polls predict that candidate A will win about 51% of the vote, while candidate B will win about 49%).*

    (a) *Suppose ther are $N = 5000$ voters in the election and that every voter in the election votes randomly and independently with those probabilities: 0.51 for candidate A and 0.49 for candidate B. Give an expression (involving a sum) for the probability that candidate A wins the election (gets more than $N/2 = 2500$ votes).*

    X is the number of votes candidate A receives
    p_A is the probability a voter votes for candidate A

    $$X \sim Bin(n, p) \text{ where } n = N = 5000, p = p_A = .51$$

    $$P(\text{A wins}) = P(X > \frac{N}{2}) = 1 - \sum_{i=0}^{\frac{N}{2}} \binom{n}{i} p^i (1-p)^{n-i} = 1 - \sum_{i=0}^{2500} \binom{5000}{i} .51^i .49^{5000-i}$$

    (b) *Compute the numerical value of this probability.*

    $$= 9.19286 \times 10^{-1}$$

    (c) *Compute the numerical value of the same probability using a Poisson approximation. Is Poisson a good approximation here?*

    X is the number of votes candidate A receives
    p_A is the probability a voter votes for candidate A

    $$X \sim Poi(\lambda) \text{ where } \lambda = N * p_A = 2550$$

    $$P(\text{A wins}) = P(X > \frac{N}{2}) = 1 - \sum_{i=0}^{\frac{N}{2}} \frac{\lambda^i}{i!} e^{-\lambda}$$

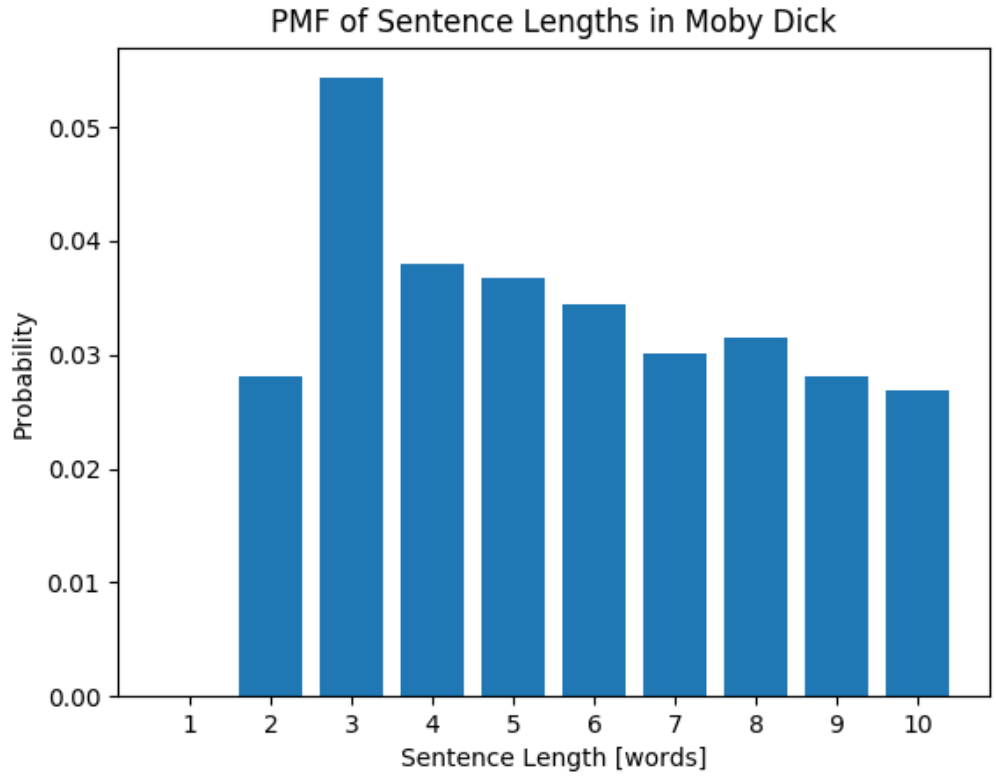
    $$= 8.36490 \times 10^{-1}$$

    The Poisson distribution is not a good approximation because the probability value (p_A) is not sufficiently small.

11. *In this problem you'll explore one (rather silly) way of trying to get a computer program to write the next great novel.*
    In the data distribution provided on the companion page for this assignement is a copy of Herman Melville's Moby Dick, from Project Gutenberg. We've processed the file so that each line is one sentence, and all the words and punctuation are separated by exactly one space.

    (a) ***[coding]*** *Make a plot of the distribution (PMF) of sentence lengths in Moby Dick, where "length" of a sentence is the total number of words and punctuation marks it contains. In other words, if you were to choose a random sentence from the novel (with all sentences equally likely), what is the probability the sentence has exactly n words (including punctuation)? Display the probabilities*

for at least $n = 1$ to 10 in the plot, but also count the longer sentences when determining those probabilities.



(b) *Consider the following approach to generating random "sentences":*

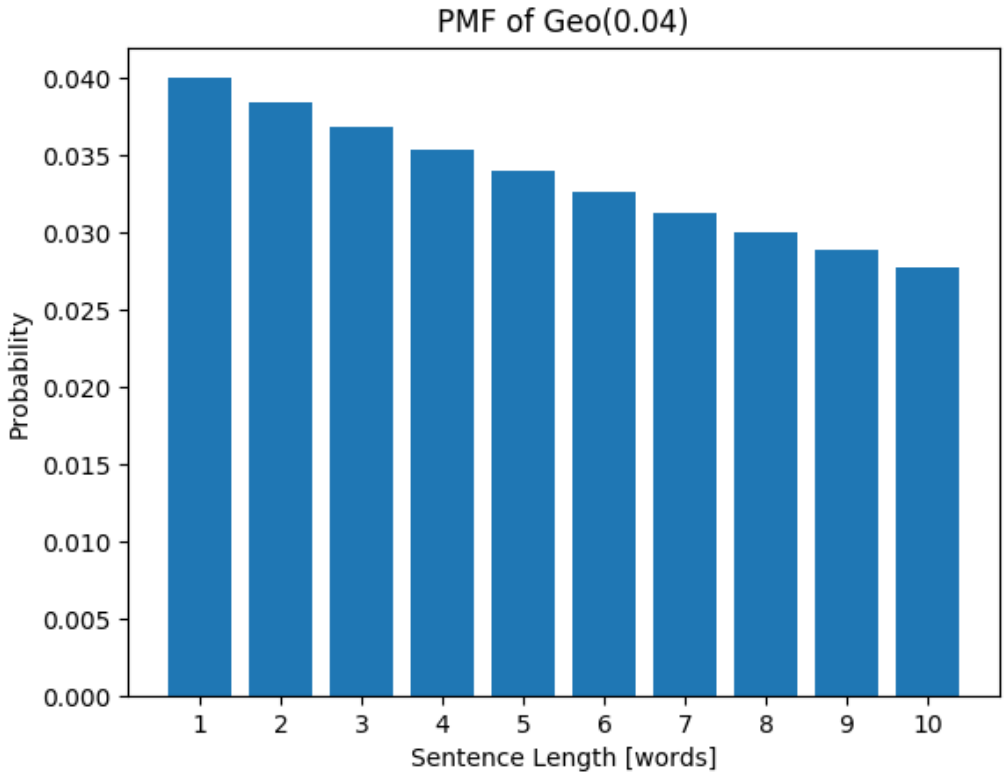
i. *Start with zero words in your sentence.*
ii. *Pick a random word from the novel. Append it to the sentence.*
iii. *With probability 0.04, stop and return the current sentence. Otherwise, go to (ii).*

Let X be the length of a sentence generated with this method. What is the distribution of X? Give the type of distribution and the values of any parameters.

This is a geometric distibution.
X is the number of words in a sentence
p is the probability that the sentence ends $= 0.04$

$$X \sim Geo(p)$$
$$P(X = n) = (1-p)^{n-1}p \text{ where } n \in \mathbb{Z}, n \geq 1$$

(c) *Plot the PMF of the distribution from part (b), from length 1 up to length 10. What is one noticeable difference between the distribution of sentence lengths in the real novel text and the distribution produced by this random-sentence generator?*



The probabilities for sentences of lengths 1 to 3 are much different. For sentence lenths 3 - 10, the probabilities are similar.