Question 1

In a particular region of the Pokemon world, there are just three types of Pokemon: Dark, Psychic, and Fighting. Kecleon, the Color Change Pokemon, has the ability to change its type to blend into its surroundings. Its goal is to become the best type. It seeks to accomplish this goal by choosing a type of Pokemon to battle uniformly at random; it may choose a Pokemon of its own type. If it wins the battle, it will keep its old type. If it loses, it will take on the type of the Pokemon that defeated it. It continues this process indefinitely.

For all questions below, assume that Dark Pokemon always win against Psychic Pokemon, Psychic Pokemon always win against Fighting Pokemon, and Fighting Pokemon always win against Dark Pokemon. If two Pokemon of the same type battle, each has an equal chance of winning.

(a) Suppose that Kecleon starts as the Fighting type. What is the probability that it will be each of the three types after two battles? (i.e. what is the probability that Kecleon will be a Fighting type after two battles, a Psychic type after two battles, and a Dark type after two battles.)

(b) After many (approaching infinity) battles, what are the steady-state fractions of battles that Kecleon spends as each of the three types?

(c) A nearby graveyard is haunted! Ghost-type Pokemon invade the population. Assume that Ghost Pokemon always lose to Dark Pokemon, and always win against Psychic Pokemon and Fighting Pokemon.

Suppose that Kecleon starts as the Fighting type when the Ghost Pokemon arrive. What is the probability that it will be each of the four types after two battles?

(d) After many (approaching infinity) battles, what are the steady-state fractions of battles that Kecleon spends as each of the four types?

Answer 1
Question 2

We have two containers $A$ and $B$. Initially we have $n$ marbles in container $A$, and 0 in container $B$. In each step, a marble is drawn uniformly at random from all the marbles, and a fair coin is flipped. The marble is then moved to the other container if the coin is heads.

(a) Define a Markov chain such that the states of the chain are the number of marbles in container $A$ at a given time.

(b) Prove that this Markov chain is aperiodic and irreducible.

(c) Let $\pi = (\pi_0, \ldots, \pi_n)$, such that $\pi_k = \binom{n}{k} \left(\frac{1}{2}\right)^n$. Prove the $\pi$ is the stationary distribution of this chain.
   
   **Hint:** Prove and use the identity $\binom{n}{k} = \frac{k+1}{n} \binom{n}{k+1} + \frac{k}{n} \binom{n}{k}$.

(d) Assume that one of the marbles is red. Since the process does not distinguish between the marbles, we can assume that if $A$ has $k$ marbles the probability that the red marble is in $A$ is $k/n$. Use this fact to show that in the stationary distribution the probability that the red marble is in $A$ is $1/2$.

   **Hint:** prove and use the identity $\frac{k}{n} \binom{n}{k} = \binom{n-1}{k-1}$.

(e) Use (d) to give an intuitive explanation to the stationary distribution in (c).

Answer 2
Question 3

Consider the printing service in the CIT building. Students and staff send printing requests to the printer, and those requests are stored in a buffer for processing. The storage capacity of the buffer is \(m\) requests: If the buffer is full, any new printing requests are discarded. We discretize time into seconds and assume that in each second, exactly one of the following events occurs:

- One new request arrives with probability \(b > 0\).
- One existing request is completed with probability \(d > 0\) if the buffer is not empty.
- No new printing requests arrive and no existing printing jobs complete. If there are between 1 and \(m - 1\) current jobs, this happens with probability \(1 - b - d\). If the buffer is full, the probability is \(1 - d\); if the buffer is empty, the probability is \(1 - b\).

The Markov chain state transition diagram in Figure 1 has states \(X_t \in \{0, 1, \ldots, m\}\) corresponding to the number of printing requests in the buffer. We define a state transition matrix \(T\) with \(m + 1\) rows and \(m + 1\) columns, and \(T[i,j] = P(X_{t+1} = j | X_t = i - 1)\). Let \(\pi_t = [\pi_{t0}, \ldots, \pi_{tm}]\), where \(\pi_{ti} \geq 0\) is the probability of being in state \(i\) at time \(t\). Note that \(\sum_i \pi_{ti} = 1\). We can recursively compute the probability of being in each state at time \(t + 1\), given the probabilities \(\pi_t\) of states at time \(t\), by applying the state transition matrix:

\[ \pi_{t+1} = \pi_t T \]

Figure 1: State transition diagram for the CIT printing service Markov chain.

For the following questions, we assume the maximum capacity of the CIT printing service is \(m = 9\), the request probability is \(b = 0.2\), and the completion probability is \(d = 0.5\).

(a) Write code to create the state transition matrix \(T\). If all states are equally probable at time 0, so \(\pi_{0,i} = \frac{1}{m+1}\), what are the probabilities \(\pi_1\) of being in each state at time 1?

(b) Apply the state transition matrix to \(\pi_0\) as defined in part (a) and compute \(\pi_1, \pi_2, \ldots, \pi_{100}\). At each iteration, also compute the magnitude of the absolute change in state probabilities:

\[ \epsilon(t) = \sum_{i=0}^{m} |\pi_{t,i} - \pi_{t-1,i}|. \]

Plot \(\epsilon(t)\) versus \(t\). Does \(\pi_t\) appear to converge to a limit? If so, report the estimated “steady-state” probabilities \(\pi_{100}\).

(c) Suppose that the queue is empty at time 0, so that \(\pi_{0,0} = 1\). Again apply the state transition matrix to compute \(\pi_1, \pi_2, \ldots, \pi_{100}\). Compute and plot the magnitude \(\epsilon(t)\) of the absolute change in state probabilities versus \(t\). From this initialization, do the state probabilities converge to the same limit as in part (b)?
(d) Suppose again that the queue is empty at time 0. An alternative way to estimate the probability of each queue state after \( n = 100 \) time steps is to use Monte Carlo simulation. Starting at \( x_0 = 0 \), write code that uses the state transition matrix to sample \( x_1 \) given \( x_0 \), \( x_2 \) given \( x_1 \), and so on until \( x_{100} \) is generated. Repeat this simulation process 1000 independent times, and set \( \tilde{\pi}_{n,i} \) to the fraction of trials in which \( x_n = i \). Compare your Monte Carlo estimates to the exact probabilities computed in part (c) by plotting the following relative error metric:

\[
\delta(i) = \frac{|\pi_n,i - \tilde{\pi}_{n,i}|}{\pi_n,i}
\]

Are the Monte Carlo estimates equally accurate for all states? Discuss why or why not. How could you improve their accuracy?

(e) Suppose that due to a mechanical problem with the printers, the job completion probability drops to \( d = 0.1 \). After 100 time steps, what is the probability that there are \( m = 9 \) jobs in the queue? What will happen to the CIT printing service? Assume that there are no jobs in the queue at \( t = 0 \).

\underline{Answer 3}
Figure 2: A small network of directed links between four webpages, and corresponding state transition probabilities. (Image courtesy of Mathematics Explorers Club, Cornell University.)

**Question 4**

When you type keywords into a search engine, it displays pages containing related terms, but how should the output be ordered? If you search for “Brown University”, you probably expect www.brown.edu to be returned as the most relevant webpage. But what if I created a new webpage that simply listed the words “Brown University” over and over, thousands of times? Would it make sense for this alternative page, which lists the Brown University name far more than Brown’s own page, to be the top ranked result?

To address the weaknesses of rankings based solely on word counts, we explore the famous pagerank algorithm, which formed the basis for at least early versions of Google’s search engine. Think of the whole internet as a directed graph where each node is a website, and there is a directed edge between node $i$ and $j$ if and only if website $i$ hyperlinks to website $j$. Intuitively, the pagerank algorithm seeks a ranking for which:

1. If a website is linked to by many other websites, then it’s an important website
2. If a website has only a few links, but those links come from authoritative sites (such as www.brown.edu), then it’s also important.
3. If a website links to a very large number of other websites, then the “importance” it transfers to each individual site is small. The pagerank algorithm uses Markov chains to allow the information provided by a link to implicitly flow both directions.

To illustrate pagerank, imagine a “random surfer” on the internet that starts at some webpage, and sequentially visits other webpages by following hyperlinks. As illustrated in Figure 2, the surfer chooses between the outgoing links from each page with equal probability. We can then define the “importance” of webpage $i$ as the long-term frequency with which this random surfer visits webpage $i$. If a node has $k$ outgoing edges, then the fraction of time a visit to this node is followed by each linked neighbor is only $\frac{1}{k}$. Denoting the state transition matrix by $T$, if the initial location of the surfer is uniform over the $m$ nodes so $\pi_0 = [\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}]$, the probability of viewing each webpage after $n$ time steps is then $\pi_0 T^n$. If $n$ is large and there are paths between all pairs of nodes, the state probabilities will converge to a stationary distribution $\pi = \pi T$. Sorting these probabilities gives the pagerank.
a) Create the state transition matrix for the small network of Figure 2. What is the equilibrium
distribution of this Markov chain? Which webpage has the highest pagerank?

![Figure 3: Networks with “deadend” sites (left) or multiple disconnected components (right)]](image)

Of course, the structure of real-world networks is more complex than Figure 2. As illustrated in
Figure 3, a naive random surfer could get stuck in a “dead end” page (an absorbing state) or
some locally connected subset of the full web. Thus, at each step, with probability $\alpha$ the surfer
“teleports” to an arbitrary website at random; each of the $m$ websites has probability $\frac{1}{m}$ of being
chosen. With probability $1 - \alpha$, the surfer follows a hyperlink as above. The overall state transition
matrix $G$ of this new Markov chain is then

$$G = (1 - \alpha)T + \alpha B, \quad B = \frac{1}{m} \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{bmatrix}$$

The matrix $G$ is also known as the “Google matrix”. In our experiments, we set $\alpha = 0.15$.

We now explore a larger network of $m = 9664$ websites. We provide the links structure of those
sites in the matrix $L$, where $L(i,j) = 1$ if there is a link from website $i$ to website $j$, and $L(i,j) = 0$
otherwise. The name variable stores the names of each website.

b) Write code that creates the state transition matrix $T$, and Google matrix $G$, for the provided
data. If website $i$ has no outgoing links, then $T(i,i) = 1$. You should double-check that for both
$T$ and $G$, the sum of the transition probabilities for each state equals 1.

c) From a uniform initial state distribution $\pi_0$, apply the Google matrix $G$ for $n = 100$ time steps to
compute $\pi_1, \pi_2, \ldots, \pi_{100}$. At each iteration $t$ also compute the magnitude of the absolute change
in state probabilities:

$$\epsilon(t) = \sum_{i=1}^{m} |\pi_t,i - \pi_{t-1},i|$$

Plot $\epsilon(t)$ versus $t$. Does $\pi_t$ appear to converge to a limit? If so, which 25 webpages have the
highest pagerank? What is the steady-state probability of visiting these top 25 sites? Discuss
your results, keeping in mind that this dataset was collected in 2002.

d) Consider a modified pagerank algorithm where instead of following “outgoing links”, the random
surfer follows “incoming links.” If the surfer is at website $i$, it chooses uniformly among the
sites that link to website $i$. This algorithm is equivalent to reversing the direction of all links
in the original data, and then applying the standard pagerank algorithm. Report the top 25
webpages, and their steady-state probabilities, for this reversed pagerank algorithm. Discuss
differences from part (c). Which ranking seems more sensible?

**Answer 4**