CS 70 Discrete Mathematics and Probability Theory
Summer 2019 James Hulett and Elizabeth Yang Midterm 2

PRINT your name: _____ _____
(First) (Last)

SIGN your name: _____

PRINT your student ID: _____

CIRCLE your exam room: VLSB 2050 Dwinelle 155 Soda 320 Soda 341A Soda 341B

Name of the person sitting to your left: _____

Name of the person sitting to your right: _____

- We will not grade anything outside of the space provided for a problem unless we are clearly told in the space provided for the question to look elsewhere.
- We will not be collecting scratch paper. Write everything you want to be graded on the exam itself.
- Assume independence means *mutual independence* unless otherwise noted.
- You may use binomial coefficients in your answers, unless the question otherwise specifies an answer form (e.g. fraction, decimal).
- Unless otherwise specified, you may use any variables from the problem in your answer.
- You may consult two handwritten double-sided sheets of notes. Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronics devices are prohibited.
- There are 14 pages (7 sheets) on the exam. Notify a proctor immediately if a page is missing.
- There are 7 questions on this exam, worth a total of 175 points.
- **You may, without proof, use theorems and facts that were proven in the notes, lecture, discussion, or homework.**
- **You have 120 minutes.**

Do not turn this page until your instructor tells you to do so.

1 True/False [3 Points Each, 30 Total]

1 point for True/False marking, 2 points for justification.

For each statement, mark whether it is true or false and give a brief justification (maximum 1 sentence, must fit in box) in the adjacent box.

(a) Working over $GF(7)$, we can use the secret-sharing scheme from class to share a secret among 7 people such that at least 3 of them must come together to recover the secret.

◯ **True**

◯ **False**

Answer: False. There are only 7 elements $GF(7)$, and we need a place to put the secret.

(b) If $\mathbb{P}[A \cap B] = \mathbb{P}[A] \times \mathbb{P}[B]$, $\mathbb{P}[A \cap C] = \mathbb{P}[A] \times \mathbb{P}[C]$, and $\mathbb{P}[B \cap C] = \mathbb{P}[B] \times \mathbb{P}[C]$, then $\mathbb{P}[A \cap B \cap C] = \mathbb{P}[A] \times \mathbb{P}[B] \times \mathbb{P}[C]$.

◯ **True**

◯ **False**

Answer: False. Pairwise independence does not imply mutual independence.

Alternative Justification: Let A and B happen independently with probability $\frac{1}{2}$ and let C be A exclusive or B.

(c) If $A \subseteq B$ and B is countable, then A is also countable.

◯ **True**

◯ **False**

Answer: True. Since $A \subseteq B$, there is a natural injection from A to B, so $|A| \leq |B| \leq |\mathbb{N}|$.

(d) If we can reduce A to the Halting Problem, A must be undecidable.

◯ **True**

◯ **False**

Answer: False. We can reduce any decidable problem to the Halting Problem by simply not using the "black box" Halting Problem solver we are given.

(e) Suppose I send $n+2k$ packets in a standard Berlekamp-Welch scheme. If there are $d < k$ corruptions, then the error polynomial $E(x)$ I solve for will only be of degree d rather than degree k.

◯ **True**

◯ **False**

Answer: False. The error polynomial will always be degree k based on how it's defined.

(f) If A is uncountable and B is uncountable, then $A - B$ is countable.

◯ **True**

◯ **False**

Answer: False. Let $A = \mathbb{R}$ and let $B = \mathbb{R}[0,1]$, then $A - B$ is all the reals not between 0 and 1 which is also uncountable.

(g) The set of recursively enumerable problems is countable.

◯ **True**

◯ **False**

Answer: True. We can map each RE problem to its recognizer, giving an injection to the set of programs, which is countable.

(h) If the error polynomial $E(x)$ from the Berlekamp-Welch scheme has fewer than k distinct roots, then that means fewer than k packets were corrupted.

◯ **True**

◯ **False**

Answer: True. Each corruption would correspond to a unique root for $E(x)$.

(i) If we want a secret sharing scheme with n participants, and we need at least k of them to unlock the secret, then we can use a polynomial of degree k.

◯ **True**

◯ **False**

Answer: False. A degree k polynomial forces a group of size at least $k+1$ to uncover the secret.

(j) Let x_1, x_2, and x_3 be three random numbers drawn with replacement from $\{1,2,...,70\}$. If A is the event that $x_1 = x_2$ and B is the event that $x_2 = x_3$, then A and B are independent of each other.

◯ **True**

◯ **False**

Answer: True. The event $A \cap B$ is just the event that $x_1 = x_2 = x_3$, which happens with probability $\frac{70}{70^3} = \frac{1}{70^2}$. Hence, $\mathbb{P}(A \cap B) = \frac{1}{70^2} = \frac{1}{70} \cdot \frac{1}{70} = \mathbb{P}(A) \cdot \mathbb{P}(B)$.

2 Short Answer [3 Points Each, 48 Total]

(a) Suppose I have a deck of 52 cards and I lost 5 cards in the deck because I was careless. I shuffle the deck and take the top card. What is the probability that the card is a spade?

Answer: By symmetry it is $\frac{1}{4}$. Because we don't know anything about the lost cards, the probability should not change.

(b) Suppose I roll two 20-sided dice. What is the probability that at least one of the dice is at least 3?

Answer: $\frac{99}{100}$. The event where at least one of the dice is at least 3 is the complement of the event where both dice show 2 or less. The probability that one of the dice rolls 2 or less is $\frac{2}{20} = \frac{1}{10}$, and since both dice are independent, the probability that both dice roll 2 or less is $\frac{1}{10} \cdot \frac{1}{10} = \frac{1}{100}$. Thus, the probability that at least one of the dice is at least 3 is $1 - \frac{1}{100} = \frac{99}{100}$.

(c) Find the number of non-negative integer solutions to $x_1 + x_2 + x_3 = 30$ where we have that at least one $x_i \leq 5$.

Answer: $\binom{32}{2} - \binom{14}{2} = 405$. We proceed by finding the total number of non-negative integer solutions, and subtract the number of non-negative integer solutions where all $x_i \geq 6$, which is the complement of the event we are trying to count.

To find the total number of non-negative integer solutions, we model using stars and bars, with 30 stars and 2 bars, giving $\binom{30+2}{2} = \binom{32}{2}$ solutions. Now, to find the solutions where each $x_i \geq 6$, we can use a change of variables: define $x_i' = x_i - 6$; now the only constraint on x_i' is that it is non-negative (another way to think of x_i' is that it is the "overflow" from 6, since we know each of our bins has at least 6 balls). Then, we have $x_1 + x_2 + x_3 = 30 \implies (x_1 - 6) + (x_2 - 6) + (x_3 - 6) = 30 - 18 = 12 \implies x_1' + x_2' + x_3' = 12$, which has $\binom{12+2}{2} = \binom{14}{2}$ solutions.

Our final answer is the difference between these two quantities, $\binom{32}{2} - \binom{14}{2} = 405$.

(d) James has 5 pairs of red socks and 10 pairs of blue socks, for a total of 30 socks. If James randomly picks 3 socks, what is the probability that he selected at least two blue socks?

Answer: $\frac{\binom{20}{3}+\binom{20}{2}\binom{10}{1}}{\binom{30}{3}}$. Denominator is all the ways he can pick 3 socks from 30. Numerator is all the ways he can pick a pair of blue socks. There are 2 cases: Either he picks 2 blue and 1 red, or 3 blue. There are $\binom{20}{3}$ ways to pick 3 blue socks, and there are $\binom{20}{2} * \binom{10}{1}$ ways to pick 2 blue socks and 1 red sock. Therefore, dividing the two gives us the answer.

(e) Suppose we want to send n packets, and we know that our channel drops a fraction p of our packets, where $0 < p < 1$. Using the Reed-Solomon encoding from class, how many *total* packets should we send?

Answer: In order for our encoding to work, we need to have that the number of packets dropped is at most the number of extra packets sent. Letting k be the number of extra packets, we need $p(n+k) = k$. Solving, we get $k = \frac{np}{1-p}$, so we want to send a total of $n + \frac{np}{1-p} = \frac{n}{1-p}$ packets.

(Technically, we should take the ceiling of this number, but that's less important.)

(f) How many length-15 bit strings with exactly 5 ones are there such that none of the ones are adjacent to each other? (*Hint: Try to relate these strings to instances of stars and bars.*)

Answer: $\binom{11}{5}$. Let zeros correspond to stars and ones correspond to bars, so we start with 10 stars and 5 bars. Since there are 5 ones, they implicitly define 6 buckets in which we can place stars. If the ones are non-consecutive, that means we must have at least 1 star in the middle 4 buckets. We can fix 4 of the stars in these 4 buckets. This leaves 6 stars, and still, 5 bars. There are $\binom{11}{5}$ configurations with 6 stars and 5 bars.

(g) Within a student club of n (distinguishable) members, there are 4 (distinguishable) committees.

(i) We ask each club member to join *exactly* 2 committees. If there are n members, how many ways can these committees be formed?

Answer: 6^n. For each member, there are $\binom{4}{2} = 6$ ways they can pick two committees to join.

(ii) We changed the rules so that all members can participate in at most 1 committee, but may also participate in none. We now want all 4 committees to have exactly 3 distinct members. How many ways can we form the committees now? Assume $n \geq 12$.

Answer: $\binom{n}{3}\binom{n-3}{3}\binom{n-6}{3}\binom{n-9}{3}$. There are $\binom{n}{3}$ ways to pick members for the first committee. Once we've done that, we have to choose 3 of the $n-3$ remaining members for the second committee, 3 of the remaining $n-6$ for the third, and 3 of the remaining $n-9$ for the fourth.

(h) Suppose Alice wishes to send Bob a length-n message $m_1, m_2, ..., m_n$, using the Berlekamp-Welch algorithm to protect against up to k corruptions. Assume that they are working over $GF(p)$ for some sufficiently large prime p.

(i) Suppose the channel corrupts exactly k packets, meaning that k of the packets Bob receives have a different value from what Alice sent. Given some specific message Alice wants to send, how many possible sequences of $n+2k$ values could Bob receive?

Answer: $\binom{n+2k}{k}(p-1)^k$ Pick the k packets that get corrupted. Each packet has $p-1$ choices because they must differ from the intended values.

(ii) If the channel randomly chooses k distinct packets to corrupt, what is the probability that none of the k corruptions occurs in the first n packets?

Answer: $\frac{\binom{2k}{k}}{\binom{n+2k}{k}}$. There are $\binom{n+2k}{k}$ possible sets of k packets that might be chosen, of which $\binom{2k}{k}$ use only the final $2k$ packets.

(iii) Suppose the channel only corrupts the first d packets, where $d < k$. How many possible polynomials could Bob get for $E(x)$? That is, how many degree exactly k polynomials $E(x)$ are there such that **(a)** $E(1) = E(2) = ... = E(d) = 0$ and **(b)** the coefficient on the x^k term is 1?

Answer: p^{k-d}. In order for both (1) and (2) to hold, $E(x)$ must be of the form $(x-1)(x-2)...(x-d)E'(x)$, where $E'(x)$ is a polynomial of degree $k-d$ such that the coefficient on the x^{k-d} term is a 1. In order to define $E'(x)$, it suffices to decide the coefficients for its remaining $k-d$ terms; there are p possibilities for each, giving us a total of p^{k-d} possible polynomials $E'(x)$ might be.

(i) Yaxin downloads a new spam filter for her email. If the filter sees an actual spam email, it sends it to the spam folder with probability 0.8. If the filter sees a non-spam email, it sends it to the spam folder with probability 0.1. For any of Yaxin's incoming email, there is a 0.2 probability it is spam. Given that Yaxin sees a message in her spam folder, what is the probability that it is not actually spam? Give your answer as a simplified fraction.

Answer: $\frac{1}{3}$. Let S be the event that a message is spam, and let F be the event that a message gets filtered to the spam folder. We want to compute $\mathbb{P}[\overline{S}|F]$. To do this, apply Bayes' Rule:

$$\begin{aligned}\mathbb{P}[\overline{S}|F] &= \frac{\mathbb{P}[F|\overline{S}]\cdot\mathbb{P}[\overline{S}]}{\mathbb{P}[F|\overline{S}]\cdot\mathbb{P}[\overline{S}]+\mathbb{P}[F|S]\cdot\mathbb{P}[S]}\\ &= \frac{0.1\cdot 0.8}{0.1\cdot 0.8+0.8\cdot 0.2}\\ &= \frac{0.08}{0.24} = \frac{1}{3}\end{aligned}$$

(j) For this question, you may use factorials and fractions.

(i) How many ways can we rearrange the 9 character string "GOBEARS!!"?

Answer: $\frac{9!}{2!}$. If we assume all characters are different, there are 7! anagrams; however, we need to divide by the number of repeats from the two exclamation points.

(ii) How many ways can we rearrange the 11 character string "BOOSTANFURD" so that all of the vowels (A, E, I, O, U) appear next to each other?

Answer: $\frac{8!4!}{2!}$. If we treat the string "AOOU" as its own letter, there are 8! anagrams. However, we can rearrange the vowels within their block. There are $\frac{4!}{2!}$ ways to do this, as the two Os are indistinguishable, giving us a total of $\frac{8!4!}{2!}$ rearrangements.

(k) Each day is sunny with probability $\frac{4}{5}$ and cloudy with probability $\frac{1}{5}$. The weather each day is independent of every other day. On a sunny day, Elizabeth goes to Cheeseboard with probability $\frac{1}{2}$. On a cloudy day she goes with probability $\frac{1}{4}$. What is the probability that she goes to Cheeseboard both next Monday and Tuesday? Leave your answer as a fraction.

Answer: $\frac{81}{400}$. For a single day, we can compute the probability she goes to Cheeseboard using the Total Probability Rule. Let C be the event she goes to Cheeseboard, and let S be the event that it is a

sunny day:

$$\begin{aligned}\mathbb{P}[C] &= \mathbb{P}[C|S]\cdot\mathbb{P}[S] + \mathbb{P}[C|\overline{S}]\cdot\mathbb{P}[\overline{S}] \\ &= \frac{1}{2}\cdot\frac{4}{5} + \frac{1}{4}\cdot\frac{1}{5} = \frac{9}{20}\end{aligned}$$

Since days are independent, our answer will be $\frac{81}{400}$.

(l) A dormitory has $n \geq 4$ students, all of whom like to gossip. One of the students hears a rumor, and tells it to one of the other $n-1$ students picked at random. After that, each student who hears the rumor tells it to another student picked uniformly at random, excluding themselves and the student who just told them the rumor. Let p_r be the probability that the rumor is told at least r times without coming back to a student who has already heard it.

Derive a formula for p_r in terms of r. Assume $3 \leq r \leq n-1$.

Answer: $\prod_{i=2}^{i=r} \frac{n-i}{n-2}$. Or $\frac{n-1}{n-1} * \frac{n-2}{n-2} * \frac{n-3}{n-2} * \frac{n-4}{n-2} * \ldots * \frac{n-r}{n-2}$. This is because the first student can choose any of the other $n-1$ students, the second student can choose from $n-2$ students (not himself/herself and the person he heard from). The third student can choose from $n-3$ students to tell out of the $n-2$ students so that no one who already heard it will hear it again. This process goes on.

3 Astronaut Asli's Anonymous Adventure [2/2/2/2/3/4 Points, 15 Total]

In Seventylandia, 70 officials are voting on whether to let Asli go to space. She needs all officials to vote "yes" in order to go. The officials wish to vote using an *anonymous* secret-sharing scheme, meaning that if Asli doesn't get a unanimous vote, she cannot tell who voted against her.

Working in $GF(71)$, we pick a degree d polynomial $P(x)$ and give official i the point $(i, P(i))$. Asli passes the vote if she can recover $P(x)$.

(a) If official i wants to vote for Asli, what should they do?

Answer: Each official i should send the point $(i, P(i))$

(b) If official i does not want to vote for her, what should they do?

Answer: The official i should send the point $(i, P(i) + 1 \bmod 71)$. In fact, it doesn't actually matter what point they should send, as long as it is not $P(i)$.

(c) What should the degree d be in order to make this scheme work?

Answer: The degree should be at most 69, so 70 correct points are needed to successfully interpolate it.

(d) Briefly explain why your scheme lets Asli recover $P(x)$ if the vote is unanimously yes.

Answer: If everyone provides the right points, then we have 70 points to interpolate a polynomial of degree at most 69.

(e) Briefly explain why Asli cannot recover $P(x)$ if the vote is not unanimously yes.

Answer: If even one person provides a different point from the original polynomial, Asli will recover a polynomial that differs from the original on at least one point, hence it is the incorrect polynomial.

(f) Briefly explain why this scheme is anonymous.

Answer: If Asli does not get the correct polynomial, any guess she has about which officials gave the wrong point and what their original points were would be consistent with some degree (at most) 69 polynomial. Thus, without knowledge of the correct polynomial, she has no reason to believe one such guess over another.

4 Can You Count It? [5 Points Each, 15 Total]

For each set below, mark whether it is countable or uncountable and prove your claim. One point for the correct bubble, 4 points for the proof.

(a) The set of finite-length strings made of lower-case English letters.

◯ **Countable**

◯ **Uncountable**

Answer: Countable. We can enumerate this set by first listing all 26 strings with one letter, then all 26^2 strings with two letters, all 26^3 strings with three letters, and so on. Any finite-length string will be listed within a finite amount of time, so this is a valid enumeration.

(b) The set of pairs of pairs of naturals: $(\mathbb{N} \times \mathbb{N}) \times (\mathbb{N} \times \mathbb{N}) = \{((n_1, n_2), (n_3, n_4)) \mid n_1, n_2, n_3, n_4 \in \mathbb{N}\}$.

◯ **Countable**

◯ **Uncountable**

Answer: Countable. We proved in lecture that the set of pairs of integers ($\mathbb{Z} \times \mathbb{Z}$) is countably infinite; since the set of pairs of natural numbers ($\mathbb{N} \times \mathbb{N}$) is an infinite subset of $\mathbb{Z} \times \mathbb{Z}$, we have that $\mathbb{N} \times \mathbb{N}$ is also countably infinite. This gives us a bijection $b : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, which we can use to create a bijection $b' : (\mathbb{N} \times \mathbb{N}) \times (\mathbb{N} \times \mathbb{N}) \to \mathbb{N} \times \mathbb{N}$ by $b'((p_1, p_2)) = (b(p_1), b(p_2))$. Hence, we have that our set is the same size as $\mathbb{N} \times \mathbb{N}$, and hence is the same size as $\mathbb{N}$.

(c) The set of functions from $\mathbb{N}$ to $\mathbb{N}$ such that $f(n) \neq n$ for all n.

◯ **Countable**

◯ **Uncountable**

Answer: Uncountable. You can do a Cantor's Diagonalization argument here similar to that in discussion. Assume for contradiction that the set is countable. We can list out all the function f_i and all the inputs x_i. Each entry (i, j) in the table is $f_i(x_j)$. We construct a new function g such that $g(x_j) = x_j + f_j(x_j) + 1$. This ensures $g(x_j) \neq x_j$ for all x_j and it ensures that $g(x_j)$ differs from $f_j(x_j)$ for all x_j. Hence, we have constructed a new function g that is not in our original listing of functions. Therefore, the set is not countable.

5 Recursive Enumerability Is In Scope. [5/3/3/5/3/3/3 Points, 25 Total]

For parts asking you to fill in the description of a program, you can either write pseudocode or describe what the program is doing in English.

A "halting converter" for a problem A is a program `C` that takes an instance of A as input and:

- If the correct answer for `x` is true, `C(x)` outputs a pair `(P,y)` such that `P(y)` halts.
- If the correct answer for `x` is false, `C(x)` outputs a pair `(P,y)` such that `P(y)` loops forever.

(a) We first prove that if A has a halting converter, A is recognizable.

(i) Suppose we have a program `C` that is a halting converter for A. Fill in the description of `R` such that it is a recognizer for A.

```
R(x):
```

Answer: We can write the following pseudocode:

```
R(x):
    (P,y) = C(x)
    P(y)
    return true
```

(ii) Prove that if the correct answer for `x` is true, `R(x)` will return true in finite time.

Answer: If the correct answer for `x` is true, the definition of a halting converter tells us that `P(y)` will halt. Hence, the first two lines of `R(x)` will eventually finish, at which point it will return true.

(iii) Prove that if the correct answer for `x` is false, `R(x)` will return false or loop forever.

Answer: If the correct answer for `x` is false, the definition of a halting converter tells us that `P(y)` will loop. Since `R(x)` calls `P(y)`, it too will loop forever.

(b) We next prove the converse: if A is recognizable, A has a halting converter.

(i) Suppose we have a recognizer `R` for A. Fill in the description of `P` such that, for an instance `x` of the problem A, `P(x)` halts if and only if the correct answer for `x` is true.

```
def P(x):
```

Answer: We can give the following pseudocode for `P`:

```
def P(x):
    retval = R(x)
    if retval = true:  halt
    else:  loop forever
```

(ii) Prove that if the correct answer for `x` is true, `P(x)` halts.

Answer: If the correct answer for `x` is true, we know from the definition of a recognizer that `R(x)` will return true in finite time. Hence, `P(x)` will take the "if" statement and immediately halt.

(iii) Prove that if the correct answer for `x` is false, `P(x)` loops forever.

Answer: If the correct answer for `x` is false, the recognizer `R` must either return false or loop forever. In the former case, `P(x)` will get to the "else" line and loop forever. In the former case, `P(x)` will get stuck at the point where it makes a call to `R(x)`, and so will again loop forever. In either case, we have that `P(x)` loops, as desired.

(iv) Fill in the description of `C` below such that it is a halting converter for A. You may use the program `P` from part (i), even if you did not complete that part.

```
def C(x):
```

Answer:

We can give the following psuedocode for `C`:

```
def C(x):
    return (P, x)
```

From the previous parts, we know that `P(x)` will halt if the correct answer for `x` is true and loop otherwise, so outputting `(P, x)` satisfies the definition of a halting converter.

6 Can You Count It? 2: Electric Boogaloo [3/5/3/3/4 Points, 18 Total]

(a) I have a group of n people. I want to choose some of them to be on a basketball team and some to be on a soccer team such that no one is on both teams and the sizes of the two teams add up to p.

(i) If I know I want k players on the basketball team, how many ways can I pick the two teams? Place your response in the box. No justification required.

Answer: Solution 1: $\binom{n}{k}\binom{n-k}{p-k}$. There are $\binom{n}{k}$ ways to fill the first team. There are $(n-k)$ people left, and I need $(p-k)$ for the second team, so there are $\binom{n-k}{p-k}$ ways to do this.

Solution 2: $\binom{n}{p}\binom{p}{k}$. First, we choose the p people to be on one of the teams in general, and from those p, we choose k to be on the basketball team. The remaining $p-k$ are automatically on the soccer team.

(ii) Let A_k be the answer you obtained above. Fill in the box such that the below identity holds, and **provide a combinatorial proof** for the identity. *You may not cite any results from lecture for this part–anything you use must be reproven.* Your answer can be in terms of p only.

$$\sum_{k=0}^{p} A_k = \binom{n}{p} \times$$

Answer: $\sum_{k=0}^{p} A_k = \binom{n}{p} \cdot 2^p$. The LHS counts (via cases) the number of ways to populate the two teams using p people total. The RHS first chooses a set of p people that are on *any* team, which explains the $\binom{n}{p}$, and decides which team each of those p people are on. For each person, there are two teams to choose from, giving 2^p.

(b) Now, let's count quaternary digit strings, i.e. strings of numbers where each digit can only be $0, 1, 2$, or 3. For example, 01312 is a quaternary string with 5 digits.

(i) How many n digit quaternary digit strings are there? Place your answer in the box.

Answer: 4^n strings, since there are 4 choices for each of the n digits.

(ii) How many n digit quaternary digit strings contain exactly k 3's? Place your answer in the box.

Answer: $3^{n-k}\binom{n}{k}$ strings. The $\binom{n}{k}$ counts all possible placement of 3's. The 3^{n-k} counts all possibilities for the non-3 bits (each can be $0, 1$, or 2).

(iii) Let Q be the answer obtained from Part (i), and T_k be the answer from Part (ii). In the box, write an expression for Q that is only in terms of T_k. **Provide a brief explanation for your answer.**

Answer: $\sum_{k=0}^{n} T_k$. The set of quaternary strings can be split based on how many 3's they contain. There could be $0, 1, 2, \ldots n$ threes. There are T_k strings for each case.

7 Streaming Services Battle (Every Day I'm Shufflin') [4 Each, 24 Total]

Write answers in the box. Any correct answer will receive full credit. However, partial credit may be awarded if sufficient work is shown.

(a) Given a playlist, the shuffle feature on Apple Music will play songs as a series of independent *shuffle cycles*. In each shuffle cycle, all songs in the list will be reordered, with each ordering equally likely. For instance, for a playlist of four songs a, b, c, d, one possible sequence of plays could be

$$a\ b\ c\ d\ |\ b\ d\ c\ a\ |\ d\ a\ c\ b\ |\ \ldots$$

where we use | to separate the shuffle cycles.

Suppose I have an Apple Music playlist with **exactly two songs**, a and b. I have this playlist on shuffle while I'm away, so when I return, I could be at any position within a shuffle cycle with equal probability. When I return, a is playing.

(i) What is the probability that the next song is b?

Answer: $\frac{3}{4}$. We have two cases: I returned at the start of a shuffle cycle (denoted by event S), or we're at the second song of a shuffle cycle (denoted by $\overline{S}$). These events each occur with probability $\frac{1}{2}$. Let N be the event that the next song is b. By the total probability rule:

$$\begin{aligned}\mathbb{P}[N] &= \mathbb{P}[N|S]\cdot\mathbb{P}[S] + \mathbb{P}[N|\overline{S}]\cdot\mathbb{P}[\overline{S}] \\ &= 1\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2} = \frac{3}{4}\end{aligned}$$

(ii) The next song played happened to be b. What is the probability that when I returned (i.e. when a was playing), it was the start of a shuffle cycle?

Answer: $\frac{2}{3}$. We apply Bayes' Rule. Again, let S be the event that I returned at the start of a shuffle cycle, and let N be the event that next song is b.

$$\begin{aligned}\mathbb{P}[S|N] &= \frac{\mathbb{P}[S\cap N]}{\mathbb{P}[N|S]\cdot\mathbb{P}[S] + \mathbb{P}[N|\overline{S}]\cdot\mathbb{P}[\overline{S}]} \\ &= \frac{\frac{1}{2}}{\frac{3}{4}} = \frac{2}{3}\end{aligned}$$

The denominator was computed in the previous section. The numerator can be computed by noting $\mathbb{P}[S\cap N] = \mathbb{P}[S] = \frac{1}{2}$.

(b) Spotify's shuffle feature works a little differently. It instead selects any copy of any song from the playlist uniformly at random to play each time. I have a Spotify playlist with 5 copies of song a, 3 copies of song b, and 2 copies of song c (10 copies total).

(i) I shuffle my Spotify playlist for 6 song plays. If I *ignore their play order*, how many different sets of 6 plays could I have gotten? Give your answer as an integer.

Answer: Since songs are now allowed to be repeated at any time, and we do not care about their play order, we use stars and bars. Our buckets are by song, and the number of repeats are now the indistinguishable "stars." Since we have 3 songs, there are 2 bars, and since we have 6 plays, there are 6 stars. The number of permutations of 2 bars and 6 stars is $\binom{8}{2} = 28$.

(ii) What is the probability that across the 6 songs played on my Spotify shuffle, I get song a twice, song b twice, and song c twice? (*You may leave your answer unsimplified.*)

Answer: Now, we do care about play order. There are $\frac{6!}{2!2!2!} = 90$ possible play orders that result in two of each song. Each play order has a $0.5 \cdot 0.5 \cdot 0.3 \cdot 0.3 \cdot 0.2 \cdot 0.2 = 0.03^2 = 0.0009$ chance of occuring. By multiplying everything, we end up with a probability of 0.081.

(c) YouTube Music's (YTM) shuffle functionality is somewhere in between Apple Music's and Spotify's. Specifically, given a playlist of n songs, YTM will still play songs as a series of *independent* length-n shuffle cycles. However, each YTM cycle will behave like Apple Music's shuffle feature (from part (a)) with probability p, and behave like Spotify's shuffle feature (from part (b)) with probability $1-p$.

I have a playlist with **exactly two songs** (one copy of each), a and b. I return when a (YTM) shuffle cycle is about to begin. (*Note: Each of the following answers may be in terms of p.*)

(i) What is the probability that the first song I hear is a and the second is b?

Answer: $\frac{1+p}{4}$. Let F_A be the event that the first song is a and S_B be the event that the second song is b. Let W be the event that the shuffle is behaving like Apple Music's. Then,

$$\begin{aligned}\mathbb{P}[F_A \cap S_B] &= \mathbb{P}[F_A \cap S_B \cap W] + \mathbb{P}[F_A \cap S_B \cap \overline{W}] \\ &= \mathbb{P}[F_A \cap S_B | W] \cdot \mathbb{P}[W] + \mathbb{P}[F_A \cap S_B | \overline{W}] \cdot \mathbb{P}[\overline{W}] \\ &= \frac{1}{2} \cdot p + \frac{1}{4} \cdot (1-p) = \frac{1+p}{4}\end{aligned}$$

(ii) What is the probability that the second song I hear is b given that the first is a?

Answer: $\frac{1+p}{2}$. Define the events as before.

$$\begin{aligned}\mathbb{P}[S_B|F_A] &= \frac{\mathbb{P}[F_A \cap S_B]}{\mathbb{P}[F_A]} \\ &= \frac{\frac{1+p}{4}}{\frac{1}{2}} = \frac{1+p}{2}\end{aligned}$$

Note, $\mathbb{P}[F_A] = \frac{1}{2}$, because regardless of if the shuffle cycle behaves like Apple Music or Spotify's shuffle, there is a $\frac{1}{2}$ chance that the first song is a.