1. Consider a random walk on a path with vertices numbered $1, 2, \ldots, n$ from left to right. At each step, we flip a coin to decide which direction to walk, moving one step left or one step right with equal probability. The random walk ends when we fall off one end of the path, either by moving left from vertex 1 or by moving right from vertex n.

   (a) Prove that if we start at vertex 1, the probability that the random walk ends by falling off the *right* end of the path is exactly $1/(n+1)$.

   Solution: Let $L(n)$ be the probability of falling off the Left end of a path of length n, starting at vertex 1. This function satisfies the recurrence

   $$L(n) = \frac{1}{2} + \frac{1}{2} \cdot L(n-1) \cdot L(n)$$

   The random walk falls off the left end of $1, 2, \ldots, n$ if and only if (1) the first step is to the left, or (2) the first step is to the right, then we fall off $2, 3, \ldots, n$ to the left, and finally (recursively) we fall off $1, 2, \ldots, n$ to the left. The base case of the recurrence is $L(1) = 1/2$ (or, if you prefer, $L(0) = 0$).

   The closed-form solution $L(n) = n/(n+1)$ now follows by induction. Specifically, for any $n > 1$, the inductive hypothesis implies

   $$L(n) = \frac{1}{2} + \frac{1}{2} \cdot \frac{n-1}{n} \cdot L(n),$$

   from which $L(n) = n/(n+1)$ follows by straightforward algebra. ■

   Solution: See part (b). ■

   Rubric: 2 points = 1 for recurrence + 1 for solution. "See part (b)" is worth 2/3 of the score for part (b), ***unless*** the part (b) solution relies on part (a).

   (b) Prove that if we start at vertex k, the probability that the random walk ends by falling off the *right* end of the path is exactly $k/(n+1)$.

   Solution: Let's suppose the path includes vertices 0 and $n+1$. Let $R(n,k)$ denote the probability that our random walk visits vertex $n+1$ before it visits vertex 0, assuming we start at vertex k. We immediately have $R(n,0) = 0$ and $R(n,n+1) = 1$.

   For all $1 \le k \le n$, the rules of the random walk imply

   $$R(n,k) = \frac{1}{2}R(n,k-1) + \frac{1}{2}R(n,k+1).$$

   In other words, the probabilities $R(n,0), R(n,1), R(n,2), \ldots, R(n,n), R(n,n+1)$ define an *arithmetic sequence*; the intermediate values are evenly spaced between $R(n,0) = 0$ and $R(n,n+1) = 1$. It follows that $R(n,k) = \frac{k}{n+1}$ for all k. ■

Solution: Let's add vertices 0 and $n+1$ to the ends of our path. Let $R(n,k)$ denote the probability that our random walk visits vertex $n+1$ before it visits vertex 0, assuming we start at vertex k. I claim that $R(n,k) = \frac{k}{n+1}$ for all integers n and k such that $n > 0$ and $0 \le k \le n+1$.

Fix an arbitrary integers n and k such that $n > 0$ and $0 \le k \le n+1$. As an inductive hypothesis, assume $R(m,j) = \frac{j}{m+1}$ for all positive integers m and j such that $0 < m < n$ and $0 \le j \le m+1$.

We immediately have $R(n,0) = 0$ and $R(n,n+1) = 1$, so suppose $1 \le k \le n$. Any random walk from vertex k to vertex $n+1$ must consist of a random walk from vertex k to vertex n, followed by an independent random walk from vertex n to vertex $n+1$. Thus,

$$\begin{aligned} R(n,k) &= R(n-1,k) \cdot R(n,n) \\ &= R(n-1,k) \cdot \frac{n}{n+1} && \text{[from part (a)]} \\ &= \frac{k}{n} \cdot \frac{n}{n+1} && \text{[induction hypothesis]} \\ &= \frac{k}{n+1} \end{aligned}$$

In all cases, we conclude that $R(n,k) = \frac{k}{n+1}$, as required. ■

Rubric: 3 points. A proof that relies on part (a) is worth full credit, but only if a standalone solution is given for part (a).

(c) Prove that if we start at vertex 1, the expected number of steps before the random walk ends is exactly n.

Solution: Let $S(n)$ be the expected number of steps before the random walk ends, assuming we start at vertex 1. We immediately observe that $S(0) = 0$ and $S(1) = 1$.

So assume $n \ge 2$. In the first step, either the random walk ends immediately, or it enters the interior path from 2 to $n-1$. In the latter case, the random walk eventually leaves this shorter path, after which we are once again at the end of a path of length n. Linearity of expectation now implies

$$S(n) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (1 + S(n-2) + S(n))$$

or equivalently, $S(n) = S(n-2) + 2$. The closed form $S(n) = n$ follows immediately by induction. ■

Solution: See part (d). ■

Rubric: 2 points. "See part (d)" is with 2/3 of the score for part (d), ***unless*** the submitted solution to part (d) relies on part (c).

(d) What is the *exact* expected length of the random walk if we start at vertex k, as a function of n and k? Prove your result is correct. (For partial credit, give a tight Θ-bound for the case $k = (n+1)/2$, assuming n is odd.)

Solution: For all integers n and k such that $0 \le k \le n+1$, let $S(n,k)$ denote the expected number of steps for the random walk to reach either vertex 0 or vertex $n+1$, assuming we start at vertex k. For all n, we immediately have $S(n,0) = S(n,n+1) = 0$. (Alternatively, if you prefer, part (c) implies $S(n,1) = S(n,n) = n$.) If $1 \le k \le n$, linearity of expectation implies

$$S(n,k) = 1 + \frac{1}{2}S(n,k-1) + \frac{1}{2}S(n,k+1),$$

or equivalently,

$$S(n,k+1) - S(n,k) = S(n,k) - S(n,k-1) - 2.$$

It follows by induction that

$$S(n,k+1) - S(n,k) \;=\; S(n,1) - S(n,0) - 2k \;=\; n - 2k,$$

and therefore, again by induction, that

$$S(n,k) \;=\; \sum_{j=0}^{k-1}(n-2j) \;=\; kn - 2\sum_{j=0}^{k-1} j \;=\; kn - k(k-1).$$

We conclude that $\boldsymbol{S(n,k) = k(n-k+1)}$. ■

Solution: For all integers n and k such that $0 \le k \le n+1$, let $S(n,k)$ denote the expected number of steps for the random walk to reach either vertex 0 or vertex $n+1$, assuming we start at vertex k.

Let's break the random walk starting at k into two phases. The first phase ends when the walk reaches either vertex 1 or vertex n for the first time; the second phase is the rest of the walk.

The expected number of steps to reach either 1 or n from k is equal to the expected number of steps to reach either 0 or $n-1$ from $k-1$. Thus, the expected length of the first phase is exactly $S(n-2,k-1)$. The expected length of the second phase is either $S(n,1)$ or $S(n,n)$, and **part (c) implies** $S(n,1) = S(n,n) = n$. So we have a simple recurrence:

$$S(n,k) \;=\; S(n-2,k-1) + n$$

To solve the recurrence, there are two cases to consider. If $k \le n/2$, then inductively expanding the recurrence k times gives us

$$S(n,k) \;=\; S(n-2k,0) + \sum_{j=0}^{k-1}(n-2j)$$

$$= nk - 2\sum_{j=0}^{k-1} k \;=\; nk - k(k-1) \;=\; k(n-k+1)$$

On the other hand, if $k > n/2$, symmetry implies $S(n,k) = S(n,n-k+1) = (n-k+1)k$. In both cases, we conclude that $\boldsymbol{S(n,k) = k(n-k+1)}$. ■

Rubric: 3 points = 1 for exact solution + 2 for proof. A proof that refers to part (c) is worth full credit only if a standalone proof is given for part (c). A $\Theta(n^2)$ bound for the special case $k = (n+1)/2$ is worth 2 points.

2. Let $A[0..2^w-1]$ and $B[0..2^w-1]$ be arrays of independent random ℓ-bit strings, and define the hash function $h_{A,B}\colon \mathcal{U} \to [m]$ by setting

$$h_{A,B}(x,y) := A[x] \oplus B[y]$$

where $\oplus$ denotes bit-wise exclusive-or. Let $\mathcal{H}$ denote the set of all possible functions $h_{A,B}$. Filling the arrays A and B with independent random bits is equivalent to choosing a hash function $h_{A,B} \in \mathcal{H}$ uniformly at random.

(a) Prove that $\mathcal{H}$ is 2-uniform.

Solution: Let (x,y) and (x',y') be arbitrary distinct elements of $\mathcal{U}$, and let i and j be arbitrary (possibly equal) hash values. To simplify notation, we define

$$a = A[x], \quad b = B[y], \quad a' = A[x'], \quad \text{and} \quad b' = B[y'].$$

Say that a, b, a', b' are ***good*** if $a \oplus b = i$ and $a' \oplus b' = j$. We need to prove that

$$\Pr\big[a, b, a', b' \text{ are good}\big] = \frac{1}{m^2}.$$

There are three cases to consider.

- Suppose $x \neq x'$ and $y \neq y'$. Then a, b, a', b' are four distinct and therefore independent random w-bit strings. There are m^4 possible values for a, b, a', b'. If we fix a and a' arbitrarily, there is exactly one good value of b and exactly one good value of b', namely, $b = a \oplus i$ and $b' = a' \oplus j$. Thus, there are m^2 good values for a, b, a', b'. We conclude that the probability that a, b, a', b' are good is $m^2/m^4 = 1/m^2$.
- Suppose $x = x'$ and $y \neq y'$. Then $a = a'$, so there are only m^3 possible values for a, b, a', b'. If we fix $a = a'$ arbitrarily, there is exactly one good value of b and exactly one good value of b', namely, $b = a \oplus i$ and $b' = a' \oplus j$. Thus, there are m ***good*** values of a, b, a', b'. We conclude that the probability that a, b, a', b' are good is $m/m^3 = 1/m^2$.
- The final case $x \neq x'$ and $y = y'$ is symmetric with the previous case.

■

Solution: See part (b). ■

Rubric: 3 points = 1 for basic setup + 1 for each interesting case. This is more detail than necessary for full credit. This is not the only correct solution. “See part (b)” is worth 3/4 of your score for part (b).

(b) Prove that $\mathcal{H}$ is 3-uniform. *[Hint: Solve part (a) first.]*

Solution: Let (x,y), (x',y'), (x'',y'') be arbitrary distinct elements of $\mathcal{U}$, and let i,j,k be arbitrary (possibly equal) hash values. To simplify notation, we define

$$a = A[x], \quad b = B[y], \quad a' = A[x'], \quad b' = B[y'], \quad a'' = A[x''], \quad b'' = B[y''].$$

Say that a,b,a',b',a'',b'' are ***good*** if $a \oplus b = i$ and $a' \oplus b' = j$ and $a'' \oplus b'' = k$. Up to symmetry, there are three cases to consider.

- Suppose x,x',x'' are distinct. Arbitrarily fix y,y',y''. There are m^3 possible values for x,x',x'', but only one of those values is good, namely $x = y \oplus i$ and $x' = y' \oplus j$ and $x'' = y'' \oplus k$. (The case where y,y',y'' are distinct is symmetric.)
- If $x = x' = x''$, then y,y',y'' must be distinct, so we can reduce to the previous case. (The case where $y = y' = y''$ is symmetric.)
- The only remaining case (up to permuting the variable names) is $x = x' \neq x''$ and $y \neq y' = y''$. In this case, there are m^4 possible values for a,b,b',a''. If we fix a arbitrarily, the only good values of the remaining variables are $b = a \oplus i$ and $b' = a \oplus j$ and $a'' = b' \oplus k = a \oplus j \oplus k$. Thus, there are exactly m good values for a,b,b',a''.

In all cases, we conclude that $\Pr[a,b,a',b',a'',b'' \text{ are good}] = 1/m^3$. ■

Rubric: 4 points = 1 for basic setup + 1 for each case. This is not the only correct solution.

(c) Prove that $\mathcal{H}$ is ***not*** 4-uniform.

Solution: For any function $h \in \mathcal{H}$ and any w-bit strings x,y,x',y', we have

$$\begin{aligned} & h(x,y) \oplus h(x',y) \oplus h(x,y') \oplus h(x',y') \\ &\quad = A[x] \oplus B[y] \oplus A[x'] \oplus B[y] \oplus A[x] \oplus B[y'] \oplus A[x'] \oplus B[y'] \\ &\quad = A[x] \oplus A[x] \oplus A[x'] \oplus A[x'] \oplus B[y] \oplus B[y] \oplus B[y'] \oplus B[y'] \\ &\quad = 0. \end{aligned}$$

It follows that for any hash values i,j,k,l, the probability

$$\Pr\big[h(x,y) = i \wedge h(x,y') = j \wedge h(x',y) = k \wedge h(x',y') = l\big]$$

is equal to $1/m^3$ if $i \oplus j \oplus k \oplus l = 0$ and equal to zero otherwise; thus, it cannot equal $1/m^4$. ■

Rubric: 3 points. This is more detail than necessary for full credit. This is not the only correct solution.

3. Suppose we want have m tasks, and we want to assign each task to one of n servers. Assume(unrealistically) that we have access to an ***ideal random*** hash function that maps any value x to a real value $h(x) \in [0,1]$. Identifying each server and each task with a different point in the domain of the hash function, we use the hash function to map both tasks and servers randomly to the unit interval $[0,1]$. Now, each task is assigned to the first server to its right on the number line (with the interval wrapping around). When a new server is added to the system, we hash it to $[0,1]$ and move tasks to it accordingly.

   (a) Suppose a new $(n+1)$-th server is added to the system. What is the expected number of tasks that need to be reassigned? Note that the expectation is taken with respect to the random positions of all the servers and all the tasks.

   Solution: The only tasks that need to be reassigned are those assigned to the new server. Since the process is as if we map all the machines to the interval from the beginning, by symmetry, in expectation $\frac{m}{n+1}$ items are assigned to the $(n+1)$-th server, and hence the expected number of tasks that are relocated is $\frac{m}{n+1}$. ■

   (b) Show that, with high probability, no server "owns" more than an $O(\log n/n)$ fraction of the interval $[0,1]$.

   Solution: Split the unit circle into $n/(2\ln n)$ disjoint *canonical* intervals, each with length $(2\ln n)/n$. (Here $\ln n$ is standard shorthand for $\log_e n$.)

   For any canonical interval I, the probability that no server lands in I is

$$\left(1 - \frac{2\ln n}{n}\right)^n \leq \left(e^{-\frac{2\ln n}{n}}\right)^n = \frac{1}{n^2}$$

   by The World's Most Useful Inequality $1 + x \leq e^x$. The union bound implies that the probability that *at least one* canonical interval contains no servers is at most $\frac{n}{2\ln n} \cdot \frac{1}{n^2} > \frac{1}{n}$. Thus, with probability at least $1 - \frac{1}{n}$, *every* canonical interval contains *at least one* server, and therefore every server owns at most $4\ln n/n$ of the unit circle. ■

   (c) Show that if we have n servers and m items where $m \geq 1000n$, the maximum load on any server is $O(\frac{m}{n} \cdot \log n)$ with high probability.

   Solution: Part (b) implies that with high probability, each server owns at most $4\ln n/n$ of the unit circle. Let us assume that this event happens and bound the probability, over the random positions of the tasks, that the maximum load exceeds $16m\ln n/n$. Since the tasks are also distributed uniformly, the expected number of tasks in any interval of size $4\ln n/n$ is at most $4m\log n/n$. Because $m \geq 1000n$, Chernoff bounds imply that with probability at least $1 - n^{-100}$, the number of tasks in any interval of length $4\ln n/n$ is at most $16m\ln n/n$.

   The union bound now implies that with probability at least $1 - 1/n - n^{-100}$, the maximum load is at most $16m\ln n/n$. ■

Rubric: 10 points = 2 for part (a) + 4 for part (b) + 4 for part (c)