Complexity in Computer Science

Discrete Math Background Solutions to Exercises

Thomas Watson

February 2, 2026

Contents

Chapter B

Exercise A.1.a: True. $S \setminus T = S \cap \overline{T} = \overline{T} \cap \overline{\overline{S}} = \overline{T} \setminus \overline{S}$.

Exercise A.1.b: Not guaranteed to be true. The former contains any elements in $S \cap T$ or in $\overline{S \cup T}$, and those are not in the latter. The latter is always a subset of the former, though.

Exercise A.1.c: True. $\overline{S \setminus T} = \overline{S \cap \overline{T}} = \overline{S} \cup \overline{\overline{T}} = \overline{S} \cup T$.

Exercise A.1.d: True. Both sides equal $\emptyset$.

Exercise A.1.e: True. Both sides equal $S \cap T$.

Exercise A.1.f: Not guaranteed to be true. The latter contains any elements in S, and those are not in the former. The former is always a subset of the latter, though.

Exercise A.1.g: True. The latter equals $\overline{S} \cup \overline{T}$, and $S \setminus T \subseteq \overline{T} \subseteq \overline{S} \cup \overline{T}$, and $T \setminus S \subseteq \overline{S} \subseteq \overline{S} \cup \overline{T}$.

Exercise A.1.h: Not guaranteed to be true. If $S \subseteq T$ then there might be an element in T but not in S, which would thus be in $\overline{S}$ but not in $\overline{T}$. It is true that if $S \subseteq T$ then $\overline{T} \subseteq \overline{S}$.

Exercise A.1.i: True. The former is equivalent to $T \subseteq \overline{S}$, and the latter is equivalent to $S \subseteq \overline{T}$, and both of those are equivalent to $S \cap T = \emptyset$.

Exercise A.2.a: $\{\emptyset, \{\emptyset\}, \{\{1\}\}, \{\emptyset, \{1\}\}\}$

Exercise A.2.b: True. $S \cap T \subseteq S$ and $S \cap T \subseteq T$.

Exercise A.2.c: Not true unless $T \subseteq S$ or $S \subseteq T$, since otherwise neither $S \cup T \subseteq S$ nor $S \cup T \subseteq T$.

Exercise A.3.a: True. For both sets, (x, y) is in the set iff $x \in Q$ and $y \in R$ and $x \in S$ and $y \in T$.

Exercise A.3.b: Not guaranteed to be true. (x, y) would be in the latter set but not in the former if, for example, $x \in Q \setminus S$ and $y \in T \setminus R$. The former is always a subset of the latter, though.

Exercise A.3.c: Not guaranteed to be true. (x, y) would be in the former set but not in the latter if, for example, $x \in Q \setminus S$ and $y \in R \cap T$. The latter is always a subset of the former, though.

Exercise A.4.a:

Mux	0	1
00	0	0
01	0	1
10	1	0
11	1	1

Exercise A.4.b:

Demux	0	1
0	00	00
1	10	01

Exercise A.4.c:

i	Dec(i)
00	1000
01	0100
10	0010
11	0001

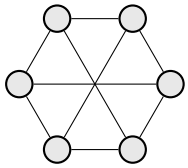
Exercise A.4.d:

x	Enc(x)
0000	undefined
0001	11
0010	10
0011	10
0100	01
0101	01
0110	01
0111	01
1000	00
1001	00
1010	00
1011	00
1100	00
1101	00
1110	00
1111	00

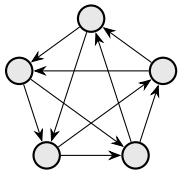
Exercise A.5:

F	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Exercise A.6.a:



Exercise A.6.b:



Exercise A.7.a: Equivalent.

P	Q	
T	T	F
T	F	T
F	T	T
F	F	F

Exercise A.7.b: Inequivalent. TT and FT are the counterexample assignments.

Exercise A.7.c: Equivalent.

P	Q	R	
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	T

Exercise A.7.d: Equivalent.

P	Q	R	
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

Exercise A.7.e: Inequivalent. TTF, TFF, FTF, FFF are the counterexample assignments.

Exercise A.8.a: True. The left side asserts $\forall x\ ((x \in R \ \vee\ x \in S) \ \Rightarrow\ x \in T)$ and the right side asserts $(\forall x\ (x \in R \ \Rightarrow\ x \in T)) \ \wedge\ (\forall x\ (x \in S \ \Rightarrow\ x \in T))$, which are equivalent by Exercise A.7.c.

Exercise A.8.b: Not guaranteed to be true. For example, it may be that $R \not\subseteq S$ and $S \not\subseteq R$ and $T = R \cap S$, in which case the left side is true but the right side is not. The right side always implies the left side, though.

Exercise A.8.c: Not guaranteed to be true. For example, it may be that $Q \not\subseteq S$ and $R = \emptyset$, in which case the left side is true (since $Q \times R = \emptyset$) but the right side is not. The right side always implies the left side, though. Also, the statement is true if Q and R are both nonempty.

Exercise A.9.a: True. Letting $y = x + 1/2$ (for example), we have $x < y < x + 1$.

Exercise A.9.b: False. Pulling the negation inside the first quantifier, $(\exists x \in \mathbb{R})\ \neg(\exists y \in \mathbb{Z})\ P(x, y)$ is true since letting x be an integer, there is no integer y such that $x < y < x + 1$.

Exercise A.9.c: True. Letting y be an integer, there is no integer x such that $x < y < x + 1$.

Exercise A.9.d: False. Pulling the negation all the way inside, $(\forall y \in \mathbb{Z})\ (\exists x \in \mathbb{R})\ P(x, y)$ is true by letting $x = y - 1/2$ (for example).

Exercise A.10.a: False. The negation is $(\forall x \in \mathbb{R})\ (\exists y \in \mathbb{R})\ (x + y \neq xy)$, which is true since if $x \neq 0$ then we can take $y = 0$, and if $x = 0$ then we can take any $y \neq 0$, and in both cases we have $x + y \neq 0 = xy$.

Exercise A.10.b: True. Taking $x = 1/2$, we have $1/2 + y \geq y$ for all y. In fact, $1/2$ is the only value of x that works here.

Exercise A.10.c: False. The negation is $(\forall x \in \mathbb{R})\ (\exists y \in \mathbb{R})\ (\lceil y \rceil \neq \lfloor y + x \rfloor)$, which is true since if $x \geq 1$ then taking $y = 0$ we have $\lceil y \rceil = 0 < 1 \leq \lfloor y + x \rfloor$, and if $x < 1$ then taking any $0 < y < 1 - x$ we have $\lceil y \rceil \geq 1 > 0 \geq \lfloor y + x \rfloor$.

Exercise A.10.d: True. If $x < 0$ then taking $y = 0$, we have that $y^2 \geq x^2$ is false, so the implication in the predicate is automatically true. If $x \geq 0$ then the implication is true for all y (in particular, for some y).

Exercise A.11.a: Equivalent. Since P doesn't depend on y and Q doesn't depend on x, both sides are equivalent to $(\exists x\ P(x)) \wedge (\forall y\ Q(y))$. Allowing x to depend on y on the right side doesn't matter. We're assuming y's domain is non-empty; if it's empty then the right side would be vacuously true while the left side could still be false.

Exercise A.11.b: Inequivalent. If the domain is $\mathbb{N}$ and $P(x, y) =$ "x is even" and $Q(x, y) =$ "x is odd" then the left side is true but the right side is false.

Exercise A.11.c: Equivalent. Both sides are always true no matter what P stands for. The left side is true since if x doesn't need to depend on y, we can allow it to depend on y but simply pick the same value of x for each value of y. The right side is true since if something holds for all y then in particular it holds for at least one y, assuming y's domain is non-empty.

Exercise A.12.a: $\exists (n_1, n_2)\, (n_1 \neq n_2 \wedge P(n_1) \wedge P(n_2) \wedge \forall m\, ((m \neq n_1 \wedge m \neq n_2) \Rightarrow \neg P(m)))$

Exercise A.12.b: $\neg\exists n\ (P(n) \iff P(n+1) \iff P(n+2))$

Exercise A.12.c: $\exists d\ \forall m\ \exists (n_1, n_2)\ (n_1 < n_2\ \wedge\ n_1 \geq m\ \wedge\ n_2 \leq m + d\ \wedge\ P(n_1)\ \wedge\ P(n_2))$

Exercise A.13: We have $f \neq o(g)$ since $f(n) = g(n)$ for all even n. We have $f = O(g)$ trivially since $f(n) \leq g(n)$ for all n. We have $g \neq O(f)$ since for any $c > 0$ and any n_0, picking an odd integer $n \geq n_0$ such that $\lceil n/2 \rceil > c$, we have $g(n) = \lceil n/2 \rceil ! > c \cdot (\lceil n/2 \rceil - 1)! = c \cdot \lfloor n/2 \rfloor ! = c \cdot f(n)$. Since the factorial function is monotonically nondecreasing, so is f since $\lfloor n/2 \rfloor \leq \lfloor (n+1)/2 \rfloor$ (with equality iff n is even), and so is g since $\lceil n/2 \rceil \leq \lceil (n+1)/2 \rceil$ (with equality iff n is odd).

Exercise A.14.a: Let c, n_0 be the constants in the definition of $f = O(g)$. For all $n \geq n_0$, we have $(f(n))^a \leq (c \cdot g(n))^a = c^a \cdot (g(n))^a$ (since the power-a function is monotonically increasing), which shows that $f^a = O(g^a)$ since c^a is a constant.

Exercise A.14.b: Let c_1, n_1 be the constants in the definition of $f_1 = O(g_1)$. Consider any constant $c > 0$, and for $c_2 = c/c_1$ let n_2 be such that for all $n \geq n_2$, $f_2(n) \leq c_2 \cdot g_2(n)$. Then for all $n \geq \max(n_1, n_2)$ we have $f_1(n) \cdot f_2(n) \leq (c_1 \cdot g_1(n)) \cdot (c_2 \cdot g_2(n)) = c \cdot (g_1(n) \cdot g_2(n))$, which shows that $f_1 f_2 = o(g_1 g_2)$.

Exercise A.14.c: If $f = O_+(g)$ then $f = O_\vee(g)$ using the same c and n_0, since if either $m \geq n_0$ or $n \geq n_0$, then $m + n \geq n_0$ since m and n are nonnegative. If $f = O_\vee(g)$ via c and n_0, then $f = O_+(g)$ via c and $2n_0$, since if $m + n \geq 2n_0$ then either $m \geq n_0$ or $n \geq n_0$.

Exercise A.15.a: $77^n - 10^n - 26^n - 26^n - 15^n$

Exercise A.15.b: $77 \cdot 76^{n-1}$

Exercise A.15.c: $77 \cdot 76^{n-1} - 10 \cdot 9^{n-1} - 26 \cdot 25^{n-1} - 26 \cdot 25^{n-1} - 15 \cdot 14^{n-1}$

Exercise A.16.a: $n \cdot \binom{n}{k} / \binom{n^2}{k}$

Exercise A.16.b: $\binom{n}{k} \cdot n^k / \binom{n^2}{k}$

Exercise A.17.a: $2/6 = 1/3$ because two outcomes are doubles with sum at most 4 (namely, $(1,1)$ and $(2,2)$) while six outcomes have sum at most 4 (namely, $(1,1)$, $(1,2)$, $(1,3)$, $(2,1)$, $(2,2)$, and $(3,1)$).

Exercise A.17.b: Let A, B, C, D be the following events respectively: pick the ordinary coin, pick the double-head coin, pick the double-tail coin, observe heads. By the law of total probability:

$$\mathbf{Pr}[D] = \mathbf{Pr}[A] \cdot \mathbf{Pr}[D \mid A] + \mathbf{Pr}[B] \cdot \mathbf{Pr}[D \mid B] + \mathbf{Pr}[C] \cdot \mathbf{Pr}[D \mid C] = \tfrac{1}{3} \cdot \tfrac{1}{2} + \tfrac{1}{3} \cdot 1 + \tfrac{1}{3} \cdot 0 = 1/2$$

$$\mathbf{Pr}[B \cap D] = \mathbf{Pr}[B] \cdot \mathbf{Pr}[D \mid B] = 1/3$$

$$\mathbf{Pr}[B \mid D] = \mathbf{Pr}[B \cap D] / \mathbf{Pr}[D] = \tfrac{1}{3} / \tfrac{1}{2} = 2/3$$

Exercise A.17.c: Let A be this event, so $\overline{A}$ is the event that all k jobs get assigned to different machines from each other. The number of outcomes in the sample space is n^k (all equally likely), and $|\overline{A}| = \binom{n}{k} \cdot k!$, so $\mathbf{Pr}[A] = 1 - \mathbf{Pr}[\overline{A}] = 1 - \binom{n}{k} \cdot k!/n^k$.

Exercise A.17.d: Denote the outcome by (x, y), which is sampled uniformly from $[n] \times [n]$. The event "$x + y$ is even" is equivalent to "x and y are either both even or both odd." Since x and y are independent, we can write:

$$\begin{aligned}\mathbf{Pr}[x+y \text{ is even}] &= \mathbf{Pr}[x \text{ is even}] \cdot \mathbf{Pr}[y \text{ is even}] + \mathbf{Pr}[x \text{ is odd}] \cdot \mathbf{Pr}[y \text{ is odd}] \\ &= \begin{cases} \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} & \text{if } n \text{ is even} \\ \frac{1-1/n}{2} \cdot \frac{1-1/n}{2} + \frac{1+1/n}{2} \cdot \frac{1+1/n}{2} = \frac{1+1/n^2}{2} & \text{if } n \text{ is odd} \end{cases}\end{aligned}$$

Exercise A.18.a: $1/3$ because the four outcomes of $S = \{\text{boy}, \text{girl}\}^2$ have equal probabilities, and conditioning on the event $\big\{(\text{boy}, \text{boy}), (\text{boy}, \text{girl}), (\text{girl}, \text{boy})\big\}$, only one of these three outcomes has two boys.

Exercise A.18.b: Represent the days of the week as $\{0, 1, 2, 3, 4, 5, 6\}$, so Tuesday is 2. The 14^2 outcomes of $S = (\{\text{boy}, \text{girl}\} \times \{0, 1, 2, 3, 4, 5, 6\})^2$ have equal probabilities. The event that at least one child is a boy born on a Tuesday has 27 outcomes: 13 outcomes have $(\text{boy}, 2)$ for the first child but not the second, 13 outcomes have $(\text{boy}, 2)$ for the second child but not the first, and there is 1 more outcome $((\text{boy}, 2), (\text{boy}, 2))$. Of these 27 outcomes, 13 of them have two boys: $((\text{boy}, i), (\text{boy}, j))$ where either $i = 2$ and $j \neq 2$ (6 possibilities) or $i \neq 2$ and $j = 2$ (6 possibilities) or $i = j = 2$ (1 possibility). So the answer is $13/27$.

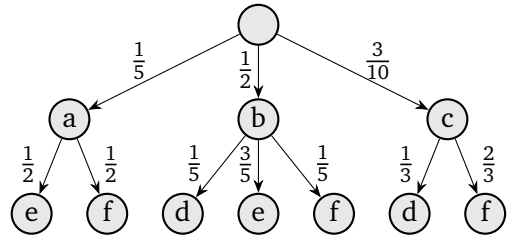
Exercise A.19: By the law of total probability, $\mathbf{Pr}[B] = \mathbf{Pr}[A] \cdot \mathbf{Pr}[B \mid A] + \mathbf{Pr}[\overline{A}] \cdot \mathbf{Pr}[B \mid \overline{A}]$. The right-side inequality follows from $\mathbf{Pr}[A] \leq 1$, $\mathbf{Pr}[B \mid A] \geq 0$, $\mathbf{Pr}[\overline{A}] \geq 0$, and $\mathbf{Pr}[B \mid \overline{A}] \leq 1$. Using $\mathbf{Pr}[\overline{A}] \cdot \mathbf{Pr}[B \mid \overline{A}] \geq 0$ we have $\mathbf{Pr}[B] \geq \mathbf{Pr}[A] \cdot \mathbf{Pr}[B \mid A] = (1 - \mathbf{Pr}[\overline{A}]) \cdot \mathbf{Pr}[B \mid A] = \mathbf{Pr}[B \mid A] - \mathbf{Pr}[\overline{A}] \cdot \mathbf{Pr}[B \mid A]$. Thus the left-side inequality follows from $\mathbf{Pr}[\overline{A}] \geq 0$ and $\mathbf{Pr}[B \mid A] \leq 1$. The above is assuming $\mathbf{Pr}[\overline{A}] > 0$. If $\mathbf{Pr}[\overline{A}] = 0$ then $\mathbf{Pr}[A] = 1$ and thus $\mathbf{Pr}[B] = \mathbf{Pr}[B \mid A]$ holds since conditioning on A doesn't change anything.

Exercise A.20:

D	d	e	f
a	0	0.1	0.1
b	0.1	0.3	0.1
c	0.1	0	0.2

D_1
0.2
0.5
0.3

D_2	0.2	0.4	0.4



Exercise A.21.a: Let random variable X be the number of runs. For $i \in [n]$ define X_i as the indicator for the i^{th} toss starting a new run, so $X = \sum_{i=1}^{n} X_i$. Note that X_1 is always 1. For $i > 1$, the i^{th} toss starts a new run when $s_{i-1}s_i \in \{\text{HT}, \text{TH}\}$, which happens with probability $p \cdot (1-p) + (1-p) \cdot p = 2p(1-p)$. Thus $\mathbf{E}[X] = \sum_{i=1}^{n} \mathbf{E}[X_i] = 1 + (n-1) \cdot 2p(1-p)$.

Exercise A.21.b: For any pair of numbers $i < j$, they will eventually get swapped with each other (exactly once) if they start in the wrong relative order (j appears earlier than i in the permutation), and they will never get swapped if they start in the correct relative order. This is because bubble sort only does adjacent swaps, so the only way to fix the relative order is for i and j to become adjacent and get swapped, and then they will never get swapped again since bubble sort only swaps inverted pairs. Letting $X_{i,j}$ be the number of times i and j are swapped with each other, we have $\mathbf{E}[X_{i,j}] = 1/2$ since the two possible relative orders are equally likely. Thus letting X be the total number of swaps, by linearity of expectation $\mathbf{E}[X] = \sum_{i<j} \mathbf{E}[X_{i,j}] = \binom{n}{2} \cdot \frac{1}{2}$.

Exercise A.22.a: By the law of total probability:

$$\begin{aligned}
\sum_{k=1}^{\infty} \mathbf{Pr}[X \geq k] &= \sum_{k=1}^{\infty} \sum_{\ell=k}^{\infty} \mathbf{Pr}[X = \ell] \\
&= \sum_{\ell=1}^{\infty} \sum_{k=1}^{\ell} \mathbf{Pr}[X = \ell] \\
&= \sum_{\ell=1}^{\infty} \mathbf{Pr}[X = \ell] \cdot \ell \\
&= \sum_{\ell=0}^{\infty} \mathbf{Pr}[X = \ell] \cdot \ell \\
&= \mathbf{E}[X]
\end{aligned}$$

Exercise A.22.b: Let X be this random variable. The event "$X \geq k$" occurs when the first $k-1$ tosses are all tails, which has probability $(1-p)^{k-1}$. So $\mathbf{E}[X] = \sum_{k=1}^{\infty} \mathbf{Pr}[X \geq k] = \sum_{k=1}^{\infty} (1-p)^{k-1}$. To evaluate this geometric series using the telescoping trick, if we multiply it by $1-(1-p) = p$ and cancel terms we get $(1-p)^0 = 1$, so dividing by p we find that the sum of the series is $1/p$.

Exercise A.23.a: This is like a geometric random variable where 6 is heads (probability $p = 1/3$) and 2 and 4 are tails, so the expectation is $1/p = 3$.

Exercise A.23.b: Let A be this event, and let s_1 denote the outcome of the first roll. By the law of total probability, $\mathbf{Pr}[A] = \mathbf{Pr}[s_1 = 6] \cdot \mathbf{Pr}[A \mid s_1 = 6] + \mathbf{Pr}[s_1 \in \{2,4\}] \cdot \mathbf{Pr}[A \mid s_1 \in \{2,4\}] + \mathbf{Pr}[s_1 \in \{1,3,5\}] \cdot \mathbf{Pr}[A \mid s_1 \in \{1,3,5\}] = \frac{1}{6} \cdot 1 + \frac{1}{3} \cdot \mathbf{Pr}[A] + \frac{1}{2} \cdot 0 = \frac{1}{6} + \frac{1}{3} \cdot \mathbf{Pr}[A]$ where $\mathbf{Pr}[A \mid s_1 \in \{2,4\}] = \mathbf{Pr}[A]$ holds because if the first roll is 2 or 4, then the rest of the experiment is distributed exactly like the original experiment (it "resets" after the first roll). Rearranging gives $\mathbf{Pr}[A] = 1/4$.

Exercise A.23.c: Let random variable X be the number of tosses, and let A be the conditioning event. For any integer $k \geq 1$, we have $\mathbf{Pr}[X = k \mid A] = \mathbf{Pr}[\text{“}X = k\text{”} \cap A]/\mathbf{Pr}[A] = \mathbf{Pr}\big[s_1 \cdots s_{k-1} \in \{2,4\}^{k-1}$ and $s_k = 6\big]/\frac{1}{4} = (\frac{1}{3})^{k-1} \cdot \frac{1}{6}/\frac{1}{4} = (\frac{1}{3})^{k-1} \cdot \frac{2}{3}$. Thus conditioned on A, X is distributed just like a geometric random variable with parameter $p = 2/3$ and thus has expectation $1/p = 3/2$.

Exercise A.24.a: $59 = 4 \cdot 13 + 7$, so the quotient is 4 and the remainder is $59 \bmod 13 = 7$. $-59 = -5 \cdot 13 + 6$, so the quotient is -5 and the remainder is $(-59) \bmod 13 = 6$.

Exercise A.24.b: First modding the individual numbers by 9 yields $(3 \cdot (4-7)) \bmod 9$. Then $(4-7) \bmod 9 = (-3) \bmod 9 = 6$, so we have $(3 \cdot 6) \bmod 9 = 18 \bmod 9 = 0$.

Exercise A.24.c:

a	0	1	2	3	4	5	6	7	8	9	10
a^{-1}		1	6	4	3	9	2	8	7	5	10

Exercise A.25.a:

x	$P(x)$
0	1
1	1
2	2

Exercise A.25.b:

Q	0	1	2
0	0	1	2
1	2	0	2
2	1	1	0

Exercise A.25.c:

$$\begin{bmatrix} (1,1)\cdot(0,2) & (1,1)\cdot(2,1) \\ (1,2)\cdot(0,2) & (1,2)\cdot(2,1) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

Exercise A.26.a: $3(x^2+1)^2+4(x^2+1)+5 = (3x^4+6x^2+3)+(4x^2+4)+5 = 3x^4+10x^2+12$

Exercise A.26.b: First, consider the case where P has only a single monomial (with nonzero coefficient) in its canonical form, say $c \cdot y^e$. Each monomial of $P(Q)$ can be obtained by picking a tuple of e many (not-necessarily-distinct) monomials from Q's canonical form

$$Q(x) = c_d \cdot x^d + c_{d-1} \cdot x^{d-1} + \cdots + c_2 \cdot x^2 + c_1 \cdot x + c_0$$

and multiplying them together, along with c. If $i_1, \ldots, i_e$ are the chosen exponents, this gives monomial $(c \cdot c_{i_1} \cdots c_{i_e}) \cdot x^{i_1 + \cdots + i_e}$, whose exponent is $\leq e \cdot d$ since $i_j \leq d$ for each $j \in [e]$. Since every resulting monomial has exponent $\leq e \cdot d$, this shows that $\deg(P(Q)) \leq e \cdot d$. Some of these monomials may cancel out (yield coefficient 0 when collected), but there is only one highest-degree monomial in the expansion (namely $(c \cdot c_d \cdots c_d) \cdot x^{d + \cdots + d}$ with nonzero coefficient and exponent $e \cdot d$, corresponding to $i_1 = \cdots = i_e = d$) so it is not cancelled out, which shows $\deg(P(Q)) = e \cdot d = \deg(P) \cdot \deg(Q)$.

Now, suppose P's canonical form has $k > 1$ many monomials, with exponents $e_k > e_{k-1} > \cdots > e_1$. Applying the previous paragraph to each of these monomials shows that $P(Q)$ is a sum of k polynomials of degrees $e_k \cdot \deg(Q)$, $\ldots$, $e_1 \cdot \deg(Q)$. Since the polynomials being summed here have distinct degrees, the degree of the sum is the largest degree of any summand: $e_k \cdot \deg(Q) = \deg(P) \cdot \deg(Q)$.

Exercise A.26.c: First, consider the case where P has only a single monomial (with nonzero coefficient) in its canonical form, say $c \cdot y_1^{e_1} \cdots y_m^{e_m}$ (and if y_i doesn't appear in the monomial, then $e_i = 0$). Then $P(Q_1, \ldots, Q_m)$ is a product of $e_1 + \cdots + e_m \leq a$ many polynomials, namely e_1 copies of Q_1, e_2 copies of Q_2, and so on. Since the total degree of a product is at most the sum of the total degrees of the factors, we have $\deg(P(Q_1, \ldots, Q_m)) \leq e_1 \cdot \deg(Q_1) + \cdots + e_m \cdot \deg(Q_m) \leq (e_1 + \cdots + e_m) \cdot b \leq a \cdot b$.

Now, suppose P's canonical form has $k > 1$ many monomials. Applying the previous paragraph to each of these monomials shows that $P(Q_1, \ldots, Q_m)$ is a sum of k polynomials each with total degree $\leq a \cdot b$, so it also has total degree $\leq a \cdot b$.

Exercise A.27: To expand $P \cdot Q$, we pick one monomial from P's canonical form and one monomial from Q's canonical form (in all possible ways) and multiply these two monomials, and sum all the resulting monomials. The total degree of a product of monomials equals the sum of their total degrees since $\left(\prod_i x_i^{a_i}\right)\left(\prod_i x_i^{b_i}\right) = \prod_i x_i^{a_i+b_i}$ has total degree $\sum_i (a_i + b_i) = \left(\sum_i a_i\right) + \left(\sum_i b_i\right)$, so this means no monomial of total degree higher than $\deg(P) + \deg(Q)$ can appear, so $\deg(P \cdot Q) \leq \deg(P) + \deg(Q)$.

The thing to worry about is that the resulting highest-total-degree monomials might all cancel each other out. Some of them may, but we show that not all of them will. Suppose $c \prod_i x_i^{d_i}$ is the monomial with lexicographically earliest exponent tuple among the highest-total-degree monomials in P's canonical form, and suppose $c' \prod_i x_i^{e_i}$ is the monomial with lexicographically earliest exponent tuple among the highest-total-degree monomials in Q's canonical form. Multiplying these yields $cc' \prod_i x_i^{d_i+e_i}$. We claim that no other product of a P monomial and a Q monomial can yield the exponent tuple $(d_1 + e_1, \ldots, d_n + e_n)$, and thus that monomial doesn't get canceled out, which shows that $\deg(P \cdot Q) = \sum_i (d_i + e_i) = \left(\sum_i d_i\right) + \left(\sum_i e_i\right) = \deg(P) + \deg(Q)$. To verify the claim, consider any highest-total-degree monomial of P, say with exponent tuple $(a_1, \ldots, a_n)$, and any highest-total-degree monomial of Q, say with exponent tuple $(b_1, \ldots, b_n)$. There are three cases:

- $(a_1, \ldots, a_n) \neq (d_1, \ldots, d_n)$ but $(b_1, \ldots, b_n) = (e_1, \ldots, e_n)$. Let i be such that $a_i \neq d_i$. The product of these monomials has exponent tuple $(a_1 + b_1, \ldots, a_n + b_n)$, which is different from $(d_1 + e_1, \ldots, d_n + e_n)$ since $a_i + b_i \neq d_i + e_i$.
- $(b_1, \ldots, b_n) \neq (e_1, \ldots, e_n)$ but $(a_1, \ldots, a_n) = (d_1, \ldots, d_n)$. This is completely analogous to the first case.
- $(a_1, \ldots, a_n) \neq (d_1, \ldots, d_n)$ and $(b_1, \ldots, b_n) \neq (e_1, \ldots, e_n)$. Let i be such that $d_i < a_i$ and $d_k = a_k$ for all $k < i$. Let j be such that $e_j < b_j$ and $e_k = b_k$ for all $k < j$. Assume that $i \leq j$ (the case $i > j$ is handled analogously). The product of these monomials has exponent tuple $(a_1 + b_1, \ldots, a_n + b_n)$, which is different from $(d_1 + e_1, \ldots, d_n + e_n)$ since $a_i > d_i$ and $b_i \geq e_i$ and thus $a_i + b_i > d_i + e_i$ (we have $b_i = e_i$ if $j > i$, and $b_i > e_i$ if $j = i$).

Exercise A.28: First, if A has size $m \times n$, then $A^\top$ has size $n \times m$, so the matrix-matrix product $A^\top A$ is defined and has size $n \times n$.

$(A^\top A)_{i,j}$ is the dot product of $A^\top$'s i^{th} row (which is A's i^{th} column) and A's j^{th} column. $(A^\top A)_{j,i}$ is the dot product of $A^\top$'s j^{th} row (which is A's j^{th} column) and A's i^{th} column. Both are the dot product of A's i^{th} and j^{th} columns, so $(A^\top A)_{i,j} = (A^\top A)_{j,i}$.

Exercise A.29.a: $v \cdot w = 1$ means there are an odd number of indices i such that $v_i = w_i = 1$. Thus for each i, this is like a biased coin toss with heads probability $p = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, and $v \cdot w = 1$ means we got an odd number of heads. The tosses are fully independent. As shown in §A.7.3, the probability that a sample from the binomial distribution $B_{n,p}$ is odd is $\frac{1}{2} - \frac{1}{2} \cdot (1-2p)^n$. With $p = \frac{1}{4}$, this gives the answer $\frac{1}{2} - \frac{1}{2^{n+1}}$.

Exercise A.29.b: Let random variable X be the number of 1s in AB, and let the indicator random variable $X_{i,j}$ be $(AB)_{i,j}$, so $X = \sum_{(i,j)\in[\ell]\times[m]} X_{i,j}$. From Exercise A.29.a, $\mathbf{E}[X_{i,j}] = \frac{1}{2} - \frac{1}{2^{n+1}}$ since $X_{i,j}$ is the dot product of A's i^{th} row with B's j^{th} column (which are uniformly random vectors of size n). By linearity of expectation, $\mathbf{E}[X] = \sum_{i,j} \mathbf{E}[X_{i,j}] = \ell \cdot m \cdot (\frac{1}{2} - \frac{1}{2^{n+1}})$. The $X_{i,j}$ random variables are not fully independent, but it doesn't matter.

Exercise A.30.a:

Adjacency matrix: $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ Transition matrix: $\begin{bmatrix} 1/2 & 1/2 & 0/2 & 0/2 \\ 1/2 & 0/2 & 0/2 & 1/2 \\ 1/5 & 2/5 & 1/5 & 1/5 \\ 0/1 & 0/1 & 1/1 & 0/1 \end{bmatrix}$

Exercise A.30.b:

$$\begin{bmatrix}0.2 & 0.1 & 0.5 & 0.2\end{bmatrix}\begin{bmatrix}0.5 & 0.5 & 0 & 0\\ 0.5 & 0 & 0 & 0.5\\ 0.2 & 0.4 & 0.2 & 0.2\\ 0 & 0 & 1 & 0\end{bmatrix} = \begin{bmatrix}0.25 & 0.3 & 0.3 & 0.15\end{bmatrix}$$

Exercise B.1.a: Let $i \in [n]$ be such that $\min(|x_1|, \ldots, |x_n|) = |x_i|$. Thus, $|x_i| \leq |x_j|$ holds for all $j \in [n]$. In particular, letting j be such that $\max(x_1, \ldots, x_n) = x_j$, we have $|x_i| \leq |x_j| = |\max(x_1, \ldots, x_n)|$. Similarly, for the other part, let $k \in [n]$ be such that $\max(|x_1|, \ldots, |x_n|) = |x_k|$. Thus, $|x_k| \geq |x_\ell|$ holds for all $\ell \in [n]$. In particular, letting ℓ be such that $\min(x_1, \ldots, x_n) = x_\ell$, we have $|x_k| \geq |x_\ell| = |\min(x_1, \ldots, x_n)|$.

Exercise B.1.b: We give a direct proof. Consider any real numbers a, b, c, d, and assume $a \geq b$ and $c \geq d$. Then $a - b \geq 0$ and $c - d \geq 0$, so $(a - b) \cdot (c - d) \geq 0$ (since nonnegative times nonnegative is nonnegative). Expanding this yields $ac + bd - ad - bc \geq 0$, which rearranges to $ac + bd \geq ad + bc$.

Exercise B.1.c: Consider any nonnegative numbers a, b, c. There are two cases.

- If $a \leq \max(b, c)$ then at least one of $\min(a, b)$ or $\min(a, c)$ equals a and the other is ≥ 0, so $\min(a, b) + \min(a, c) \geq a = \min(a, b + c)$.
- If $a > \max(b, c)$ then $\min(a, b) + \min(a, c) = b + c \geq \min(a, b + c)$.

Exercise B.2: This is false. Here's a counterexample, in which each node has indegree 1 and outdegree either 0 or 2:

Exercise B.3: Consider any $S \subseteq \{0,1\}^n$ with $|S| > 2^{n-1}$. Say S is a set of pigeons and $\{0,1\}^{n-2}$ is a set of holes, and pigeon x flies into hole $x_1x_2\cdots x_{n-2}$ (the first $n-2$ bits of x). Analogous to the pigeonhole principle, we claim that some hole must get at least 3 pigeons: If each hole got at most 2 pigeons, the number of pigeons would be at most twice the number of holes: $|S| \leq 2 \cdot 2^{n-2} = 2^{n-1}$.

Thus, there exist distinct $x, y, z \in S$ that all agree on their first $n-2$ bit positions. In particular, x, y, z all have distance ≤ 2 from each other. If $\text{dist}(x,y) = 1$ and $\text{dist}(y,z) = 1$ then y has two neighbors x and z in S. Otherwise, either $\text{dist}(x,y) = 2$ or $\text{dist}(y,z) = 2$. Suppose $\text{dist}(x,y) = 2$. (The case where $\text{dist}(y,z) = 2$ is analogous.) Then x and y disagree on both of the last 2 positions. We can't have $\text{dist}(x,z) = 2$ (since then z would equal y), and we can't have $\text{dist}(y,z) = 2$ (since then z would equal x). Thus $\text{dist}(x,z) = \text{dist}(y,z) = 1$, so z has two neighbors x and y in S.

Exercise B.4: For every nonempty $P \subseteq [n]$, we have $1 \leq \sum_{x \in P} x \leq 1 + 2 + \cdots + n \leq n^2 - 2$ (since $n \geq 4$). We view each nonempty subset of U as a pigeon, and view each element of $[n^2 - 2]$ as a hole, and have pigeon P fly into hole $\sum_{x \in P} x$. There are $2^{|U|} - 1 \geq n^2 - 1$ pigeons and only $n^2 - 2$ holes, so some hole gets two distinct pigeons $P \neq Q$, so $\sum_{x \in P} x = \sum_{x \in Q} x$. We can't have $P \subseteq Q$ (since otherwise $\sum_{x \in P} x < \sum_{x \in Q} x$) or $Q \subseteq P$ (since otherwise $\sum_{x \in Q} x < \sum_{x \in P} x$). Thus, defining $S = P \setminus Q$ and $T = Q \setminus P$, we have $S \cap T = \emptyset$ and $S \neq \emptyset$ and $T \neq \emptyset$ and $S, T \subseteq U$ and:

$$\sum_{x \in S} x \;=\; \left(\sum_{x \in P} x\right) - \left(\sum_{x \in P \cap Q} x\right) \;=\; \left(\sum_{x \in Q} x\right) - \left(\sum_{x \in P \cap Q} x\right) \;=\; \sum_{x \in T} x$$

Exercise B.5.a: Consider any such matrix M. Each column k has at least two equal entries by the pigeonhole principle (pigeon i flies into hole $M_{i,k}$; there are $n+1$ pigeons and n holes). Now, we use the pigeonhole principle differently: Each pigeon represents a column, and each hole represents a pair $(c, \{i, j\})$ where $c \in [n]$ and $\{i, j\} \subseteq [n+1]$ is a set of two row indices. For each column k, we pick two equal entries, say $M_{i,k} = M_{j,k} = c$ (if there are multiple options, it doesn't matter which two equal entries we pick), and have pigeon k fly into hole $(c, \{i, j\})$. Since there are $n\binom{n+1}{2}$ holes, which is less than the number of pigeons, some hole $(c, \{i, j\})$ must get two pigeons k and ℓ. That means $M_{i,k} = M_{j,k} = M_{i,\ell} = M_{j,\ell} = c$.

Exercise B.5.b: For each pair $(c, \{i, j\})$ where $c \in [n]$ and $\{i, j\} \subseteq [n+1]$ is a set of two row indices, create a column k where $M_{i,k} = M_{j,k} = c$ and the other $n-1$ entries of column k are the other $n-1$ values of $[n]$ besides c, in an arbitrary order. Consider any 2×2 submatrix $M_{\{i,j\},\{k,\ell\}}$. If $\{i, j\}$ is not the pair of rows associated with column k, then $M_{i,k} \neq M_{j,k}$. If $\{i, j\}$ is not the pair of rows associated with column ℓ, then $M_{i,\ell} \neq M_{j,\ell}$. If $\{i, j\}$ is associated with both columns k and ℓ, then they must have different values of c, say $M_{i,k} = M_{j,k} = c$ and $M_{i,\ell} = M_{j,\ell} = c'$ for some $c' \neq c$. In all cases, $M_{\{i,j\},\{k,\ell\}}$ is not monochromatic.

Exercise B.6: We give a direct proof. Suppose $S \subseteq \{0,1\}^n$ and $|S| = 2^{n-1}$ and for all distinct $x, y \in S$ we have $\text{dist}(x,y) > 1$.

Claim. *For all $x, y \in \{0,1\}^n$ with $\text{dist}(x,y) = 1$, exactly one of x or y is in S.*

Proof. Let $i \in [n]$ be the index where $x_i \neq y_i$. Suppose the elements of S are pigeons, and the elements of $\{0,1\}^{n-1}$ are holes, and each pigeon flies into the hole obtained by deleting the i^{th} bit. No hole gets more than one pigeon, because otherwise the two pigeons would be distinct strings in S with distance 1 (differing only in the i^{th} bit). By the pigeonhole principle, since the number of pigeons equals the number of holes, every hole must have exactly one pigeon. Now x and y can't both be in S, because they would go to the same hole. Also, at least one of x or y must be in S, because otherwise some hole would get no pigeons (since no other string can agree with x and y in all but the i^{th} bit). ■

Now, consider an arbitrary $x \in \{0,1\}^n$ and let $w = \text{weight}(x)$. Define strings called $x^0, x^1, x^2, \ldots, x^w \in \{0,1\}^n$ where $x^0 = x$, and for each $i \in [w]$, x^i is obtained from x^{i-1} by flipping a single 1 to 0. Note that $\text{dist}(x^i, x^{i-1}) = 1$, and x^w is the all-0 string. By the claim, $x^0 \in S$ iff $x^1 \notin S$ iff $x^2 \in S$ iff $x^3 \notin S$, and so on. If w is even, we have $x \in S$ iff the all-0 string is in S. If w is odd, we have $x \in S$ iff the all-0 string is not in S.

Thus, if the all-0 string is in S then S contains all even-weight strings and no odd-weight strings, and if the all-0 string is not in S then S contains all odd-weight strings and no even-weight strings.

Exercise B.7: We give a direct proof. Consider any such graph, and assume every node has degree at least $(n-1)/2$. To see that the graph is connected, consider any two nodes u and v. Now, we do a case analysis: If $\{u, v\}$ is an edge, then of course there is a walk (of length 1) between u and v. Now, assuming $\{u, v\}$ is not an edge, u's and v's neighbors must all be among the other $n-2$ nodes. Since u's neighbor set has size $\geq (n-1)/2$, which is more than half of the other $n-2$ nodes, and similarly for v's neighbor set, these two sets must intersect. That is, u and v have at least one common neighbor, say w. Then $u - w - v$ is a walk (of length 2).

Exercise B.8.a: We give a contrapositive proof. Consider any $x \in \{0,1\}^n$, and assume there do not exist two consecutive 1s in x. Then each 1 is followed by a 0, except for possibly a 1 at the end of x. Thus the number of 1s is at most one more than the number of 0s in x. In other words, $\text{weight}(x) \leq (n - \text{weight}(x)) + 1$, which rearranges to $\text{weight}(x) \leq (n+1)/2 < n/2 + 1$.

Exercise B.8.b: We give a contrapositive proof. Consider any $x \in \{0,1\}^n$, and assume x is a palindrome. If n is even then every 0 in the left half can be matched to a corresponding 0 in the right half, and similarly for 1s, so x has an even number of 0s and an even number of 1s. If n is odd and x has middle bit 1, then x has an even number of 0s (since every 0 to the left of middle can be matched to a corresponding 0 to the right of middle). Similarly, if n is odd and x has middle bit 0, then x has an even number of 1s. In all cases, x has either an even number of 0s or an even number of 1s, which means it's not true that x has an odd number of 0s and an odd number of 1s.

Exercise B.9: We give a contrapositive proof. Consider any undirected graph with $n \geq 2$ nodes, and assume there exist two such cycles. Since both cycles have more than $n/2$ nodes, there must exist some node that's on both cycles. This node has degree ≥ 4 since it's incident to two edges on one cycle and two edges on the other cycle, and these four edges are distinct since the cycles share no edges. Thus, it is not the case that every node has degree ≤ 3.

Exercise B.10: Suppose there does not exist an $i \in \{1,2\}$ such that $a_i \leq 2c_i$ and $b_i \leq 2c_i$. Pulling the negation inside, this means that either $a_1 > 2c_1$ or $b_1 > 2c_1$, and either $a_2 > 2c_2$ or $b_2 > 2c_2$. Consider two cases: either $c_1 \geq c_2$ or $c_2 \geq c_1$. Suppose $c_1 \geq c_2$. If $a_1 > 2c_1$ then $a_1 + a_2 \geq a_1 > 2c_1 \geq c_1 + c_2$. If $b_1 > 2c_1$ then $b_1 + b_2 \geq b_1 > 2c_1 \geq c_1 + c_2$. Now, suppose $c_2 \geq c_1$. If $a_2 > 2c_2$ then $a_1 + a_2 \geq a_2 > 2c_2 \geq c_1 + c_2$. If $b_2 > 2c_2$ then $b_1 + b_2 \geq b_2 > 2c_2 \geq c_1 + c_2$. In all cases, either $a_1 + a_2 > c_1 + c_2$ or $b_1 + b_2 > c_1 + c_2$, so $\max(a_1 + a_2,\ b_1 + b_2) > c_1 + c_2$.

Exercise B.11.a: We give a contrapositive proof. Suppose $> \sqrt{\varepsilon}m$ rows are *heavy*, meaning the weight is $\geq \sqrt{\varepsilon}n$. Then

$$\begin{aligned}
\text{weight}(M) &\geq \text{weight(heavy rows of } M) \\
&= \textstyle\sum_{\text{heavy row } i} \text{weight(row } i) \\
&\geq \textstyle\sum_{\text{heavy row } i} \sqrt{\varepsilon}n \\
&= (\text{number of heavy rows}) \cdot \sqrt{\varepsilon}n \\
&> \sqrt{\varepsilon}m \cdot \sqrt{\varepsilon}n \\
&= \varepsilon mn
\end{aligned}$$

where we used $\sqrt{\varepsilon}n > 0$ in the fifth line.

Exercise B.11.b: We give a contrapositive proof. Suppose $\mathbf{Pr}_{s_1 \sim D_1}[s_1 \text{ is heavy}] > \sqrt{\varepsilon}$ where *heavy* means $\mathbf{Pr}_{s_2 \sim D^{s_1}}[s \in B] \geq \sqrt{\varepsilon}$. Let $A = \{\text{all heavy } s_1\} \times S_2$ and partition A into $\bigcup_{\text{heavy } s_1} A^{s_1}$ where $A^{s_1} = \{s_1\} \times S_2$. Using the law of total probability,

$$\begin{aligned}
\mathbf{Pr}_D[B] &\geq \mathbf{Pr}_D[B \cap A] \\
&= \textstyle\sum_{\text{heavy } s_1} \mathbf{Pr}_D[A^{s_1}] \cdot \mathbf{Pr}_D\left[B \,\middle|\, A^{s_1}\right] \\
&= \textstyle\sum_{\text{heavy } s_1} \mathbf{Pr}_{D_1}[s_1] \cdot \mathbf{Pr}_{s_2 \sim D^{s_1}}[s \in B] \\
&\geq \textstyle\sum_{\text{heavy } s_1} \mathbf{Pr}_{D_1}[s_1] \cdot \sqrt{\varepsilon} \\
&= \mathbf{Pr}_{s_1 \sim D_1}[s_1 \text{ is heavy}] \cdot \sqrt{\varepsilon} \\
&> \sqrt{\varepsilon} \cdot \sqrt{\varepsilon} \\
&= \varepsilon
\end{aligned}$$

where we used $\sqrt{\varepsilon} > 0$ in the sixth line.

Exercise B.12.a: $\Leftarrow$: We give a direct proof. Suppose $1/3 \leq \mathbf{Pr}[B] \leq 1/2$. Then trivially, $1/3 \leq \mathbf{Pr}[B] \leq 2/3$, so we can let $A = B$.

$\Rightarrow$: We give a direct proof. Suppose $1/3 \leq \mathbf{Pr}[A] \leq 2/3$, and consider two cases. If $\mathbf{Pr}[A] \leq 1/2$ then we can let $B = A$. If $\mathbf{Pr}[A] > 1/2$ then $\mathbf{Pr}[\overline{A}] = 1 - \mathbf{Pr}[A] < 1 - 1/2 = 1/2$ and $\mathbf{Pr}[\overline{A}] = 1 - \mathbf{Pr}[A] \geq 1 - 2/3 = 1/3$, so we can let $B = \overline{A}$.

Exercise B.12.b: $\Rightarrow$: We give a direct proof. Suppose the support equals the whole sample space—every outcome has positive probability. Consider any event A other than the whole sample space. There exists an outcome $s \notin A$, and $\mathbf{Pr}[s] > 0$ by assumption. Thus $\mathbf{Pr}[A] \leq 1 - \mathbf{Pr}[s] < 1$.

$\Leftarrow$: We give a contrapositive proof. Suppose the support does not equal the whole sample space. Then the support itself is an event with probability 1, thus demonstrating that not every event besides the whole sample space has probability < 1.

Exercise B.13: $\Rightarrow$: We give a direct proof. Suppose $A \triangle B \subseteq C$. We claim that $A \cup C \subseteq B \cup C$: For any $x \in A \cup C$, if $x \in C$ then of course $x \in B \cup C$; if $x \in A$ then either $x \in B \subseteq B \cup C$ or $x \in A \setminus B \subseteq A \triangle B \subseteq C \subseteq B \cup C$. Similarly, $B \cup C \subseteq A \cup C$, so $A \cup C = B \cup C$.

$\Leftarrow$: We give a direct proof. Suppose $A \cup C = B \cup C$. We claim that $A \setminus B \subseteq C$: If $x \in A \setminus B$ then $x \in A \subseteq A \cup C \subseteq B \cup C$ and $x \notin B$, so $x \in C$. Similarly, $B \setminus A \subseteq C$, so $A \triangle B \subseteq C$.

Exercise B.14.a: Consider any directed graph G and node s.

$\Rightarrow$: We give a direct proof. Suppose G is strongly connected, and consider any node v. By the definition of strong connectedness, there exist a walk from s to v and a walk from v to s in G.

$\Leftarrow$: We give a direct proof. Suppose for every node v, there exist a walk from s to v and a walk from v to s in G. To see that G is strongly connected, we consider any two nodes u and v and show that there is a walk from u to v. By assumption, there exist a walk from u to s and a walk from s to v. Combining these walks produces a walk from u to v.

Exercise B.14.b: Consider any such G.

$\Rightarrow$: We give a direct proof. Suppose G is strongly connected, and consider any three distinct nodes u, v, w. By assumption, there exist a walk from u to v, a walk from v to w, and a walk from w to u. Combining these walks produces a closed walk.

$\Leftarrow$: We give a direct proof. Suppose for every three distinct nodes, there exists a closed walk containing them. To see that G is strongly connected, we consider any two nodes u and v and show that there is a walk from u to v. Let w be an arbitrary node other than u and v (which exists since G has at least three nodes). By assumption, there exists a closed walk containing u, v, w. If we begin at u and follow the closed walk, it will eventually reach v (possibly after going through w). Thus there is a walk from u to v.

Exercise B.14.c: Consider any such $G = (V, E)$.

$\Rightarrow$: We give a direct proof. Suppose G is strongly connected, and consider any partition into sets $S \neq \emptyset$ and $T \neq \emptyset$ (so $V = S \cup T$ and $S \cap T = \emptyset$). There exist nodes $s \in S$ and $t \in T$, and since G is strongly connected, there exists a walk from s to t. This walk must eventually cross from S to T, since it starts in S and ends in T. The last edge (u, v) on the walk such that $u \in S$ must have $v \in T$.

$\Leftarrow$: We give a contrapositive proof. Suppose G is not strongly connected. Then there does not exist a walk from some node s to some node t. Let S be the set of all nodes reachable from s (that is, $S = \{v \in V : \exists \text{ walk from } s \text{ to } v\}$) and let $T = V \setminus S$. Then $s \in S$ and $t \in T$, so S and T are nonempty, and they form a partition of V. We claim there does not exist an edge $(u, v) \in E$ such that $u \in S$ and $v \in T$. Suppose for contradiction there is such an edge. Then on one hand $v \notin S$ (since $v \in T$), but on the other hand $v \in S$ since there's a walk from s to u (since $u \in S$), and appending the edge (u, v) yields a walk from s to v.

Exercise B.15: Consider any real numbers a, b, c. Let $m = \min(a, b)$ and $M = \max(a, b)$.

$\Leftarrow$: We give a direct proof. Suppose there exists $0 \leq x \leq 1$ such that $c = xa + (1-x)b$. Since $x \geq 0$ and $m \leq a \leq M$, we have $xm \leq xa \leq xM$. Since $1 - x \geq 0$ and $m \leq b \leq M$, we have $(1-x)m \leq (1-x)b \leq (1-x)M$. Thus $c = xa + (1-x)b \geq xm + (1-x)m = m$ and $c = xa + (1-x)b \leq xM + (1-x)M = M$.

$\Rightarrow$: We give a direct proof. Suppose $m \leq c \leq M$. If $m = M$ then $a = b = c$ and thus $c = xa + (1-x)b$ holds for *all* x (and we only need it to hold for *some* $0 \leq x \leq 1$). Now, assume $m < M$. Letting $y = (M-c)/(M-m)$, we have $y \geq 0$ since $M - c \geq 0$, and $y \leq 1$ since $M - c \leq M - m$, and

$$\begin{aligned} ym + (1-y)M &= \frac{M-c}{M-m} \cdot m + \left(\frac{M-m}{M-m} - \frac{M-c}{M-m}\right) \cdot M \\ &= \frac{(M-c) \cdot m + (c-m) \cdot M}{M-m} \\ &= \frac{c \cdot (M-m)}{M-m} \\ &= c \end{aligned}$$

by definition. If $a < b$ so $a = m$ and $b = M$, then letting $x = y$, this means $0 \leq x \leq 1$ and $c = xa + (1-x)b$ as desired. If $a > b$ so $a = M$ and $b = m$, then letting $x = 1 - y$, we have $0 \leq x \leq 1$ and $c = ym + (1-y)M = (1-x)b + xa$ as desired.

Exercise B.16.a: $\Rightarrow$: We give a direct proof. Suppose f is surjective. For each $y \in T$, define $g(y)$ to be an arbitrary preimage of y under f. Now, for each $y \in T$, we have $(f \circ g)(y) = f(g(y)) = y$; thus $f \circ g$ is the identity.

$\Leftarrow$: We give a direct proof. Suppose there exists g such that $f \circ g$ is the identity. Now, for each $y \in T$, y has at least one preimage under f, namely $g(y)$, since $f(g(y)) = y$; thus f is surjective.

Exercise B.16.b: $\Rightarrow$: We give a direct proof. Suppose f is injective. For each $y \in T$, if y has a preimage under f then define $g(y)$ to be that unique preimage, otherwise define $g(y)$ to be an arbitrary element of S. Now, for each $x \in S$, we have $(g \circ f)(x) = g(f(x)) = x$ since x is the unique preimage of $f(x)$; thus $g \circ f$ is the identity.

$\Leftarrow$: We give a direct proof. Suppose there exists g such that $g \circ f$ is the identity. Now, for each $y \in T$, if w and x are both preimages of y under f, then we have $w = g(f(w)) = g(y) = g(f(x)) = x$, so y can't have more than one distinct preimage; thus f is injective.

Exercise B.17: $\Leftarrow$: Suppose for contradiction that n is odd and there exists a connected undirected graph with n nodes such that every node has degree exactly 3. On one hand, the sum of the degrees of all nodes is even (Theorem B.4). On the other hand, the sum of the degrees of all nodes is $3n$, which is odd since 3 and n are both odd.

$\Rightarrow$: We give a direct proof. Suppose $n \geq 4$ is even. Define an undirected graph with nodes $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$, where node v's neighbors are $(v-1) \bmod n$ and $(v+1) \bmod n$ and $(v + n/2) \bmod n$ (note that $n/2$ is an integer), which are distinct since $n \geq 4$. This is a valid undirected graph because if u is a neighbor of v then v is a neighbor of u. In more detail, $(((v-1) \bmod n)+1) \bmod n = v$ and $(((v+1) \bmod n)-1) \bmod n = v$ and $(((v+n/2) \bmod n)+n/2) \bmod n = (v+n) \bmod n = v$. By construction, every node has degree exactly 3. Furthermore, this graph is connected since for all nodes u and v, v can be reached from u by taking $(v-u) \bmod n$ many $+1$ steps.

Exercise B.18.a: $\Rightarrow$: We give a contrapositive proof. Suppose $A \cap C \neq \emptyset$ and $B \cap D \neq \emptyset$. That means there exist x and y such that $x \in A$ and $x \in C$ and $y \in B$ and $y \in D$. Thus $(x, y) \in A \times B$ and $(x, y) \in C \times D$, so $(A \times B) \cap (C \times D) \neq \emptyset$.

$\Leftarrow$: We give a contrapositive proof. Suppose $(A \times B) \cap (C \times D) \neq \emptyset$. That means there exists (x, y) such that $(x, y) \in A \times B$ and $(x, y) \in C \times D$. Thus $x \in A \cap C$ and $y \in B \cap D$, so $A \cap C \neq \emptyset$ and $B \cap D \neq \emptyset$.

Exercise B.18.b: First, we claim that $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D)$ holds no matter what. To see this, consider any $(x, y) \in (A \times B) \cup (C \times D)$. If $(x, y) \in A \times B$ then $x \in A \subseteq A \cup C$ and $y \in B \subseteq B \cup D$, so $(x, y) \in (A \cup C) \times (B \cup D)$. Analogous reasoning works if $(x, y) \in C \times D$. This proves the claim. Now, onward to the "iff":

$\Rightarrow$: We give a contrapositive proof. Suppose either $A \not\subseteq C$ and $D \not\subseteq B$, or $C \not\subseteq A$ and $B \not\subseteq D$. Assume $A \not\subseteq C$ and $D \not\subseteq B$ (analogous reasoning works if $C \not\subseteq A$ and $B \not\subseteq D$). Thus there exist $x \in A \setminus C$ and $y \in D \setminus B$. Now $(x, y) \in A \times D \subseteq (A \cup C) \times (B \cup D)$ but $(x, y) \notin (A \times B) \cup (C \times D)$: We have $(x, y) \notin A \times B$ since $y \notin B$, and $(x, y) \notin C \times D$ since $x \notin C$.

$\Leftarrow$: We give a direct proof. Suppose $A \subseteq C$ or $D \subseteq B$, and $C \subseteq A$ or $B \subseteq D$, both hold. Since we already know that $(A \times B) \cup (C \times D) \subseteq (A \cup C) \times (B \cup D)$, we just need to show that $(A \cup C) \times (B \cup D) \subseteq (A \times B) \cup (C \times D)$. Consider any $(x, y) \in (A \cup C) \times (B \cup D)$, and consider the four possibilities for our assumption:

- Suppose $A \subseteq C$ and $C \subseteq A$, so $x \in A = C$. If $y \in B$ then $(x, y) \in A \times B$. If $y \in D$ then $(x, y) \in C \times D$.
- Suppose $D \subseteq B$ and $B \subseteq D$, so $y \in B = D$. If $x \in A$ then $(x, y) \in A \times B$. If $x \in C$ then $(x, y) \in C \times D$.
- Suppose $A \subseteq C$ and $B \subseteq D$, so $x \in A \cup C = C$ and $y \in B \cup D = D$. Then $(x, y) \in C \times D$.
- Suppose $D \subseteq B$ and $C \subseteq A$, so $x \in A \cup C = A$ and $y \in B \cup D = B$. Then $(x, y) \in A \times B$.

In all cases, $(x, y) \in (A \times B) \cup (C \times D)$.

Exercise B.19.a: Note that $|S \cup T| = |S| + |T| - |S \cap T|$ since $|S| + |T|$ counts each element of $S \cup T$ once, unless the element is in $S \cap T$, in which case it is counted twice; subtracting $|S \cap T|$ corrects the overcounting. Rearranging gives $|S \cap T| = |S| + |T| - |S \cup T| \geq |S| + |T| - |U|$ since $|S \cup T| \leq |U|$.

Exercise B.19.b: Suppose for contradiction there exists a graph with n nodes such that some independent set S has size $\geq n/2+1$ and some clique T has size $\geq n/2+1$. Letting V be the set of all nodes, by Exercise B.19.a we have $|S \cap T| \geq |S|+|T|-|V| \geq (n/2+1)+(n/2+1)-n = 2$. Thus there exist distinct nodes $u, v \in S \cap T$. On one hand $\{u, v\}$ is not an edge, since $u, v \in S$ and S is an independent set. On the other hand $\{u, v\}$ is an edge, since $u, v \in T$ and T is a clique.

Exercise B.20: Suppose for contradiction there exist positive numbers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ such that for every $i \in [n]$ we have $a_i/b_i < c$, where we define $c = (a_1 + \cdots + a_n)/(b_1 + \cdots + b_n)$. Since multiplying both sides by $b_i > 0$ preserves the direction of the inequality, we have $a_i < c \cdot b_i$ for every i. Summing these inequalities over all i and using the distributive law, we get $a_1 + \cdots + a_n < c \cdot b_1 + \cdots + c \cdot b_n = c \cdot (b_1 + \cdots + b_n)$. Rearranging yields $(a_1 + \cdots + a_n)/(b_1 + \cdots + b_n) < c$ (since dividing both sides by $b_1 + \cdots + b_n > 0$ preserves the direction of the inequality). This contradicts the definition of c.

Exercise B.21.a: Consider any directed graph, and suppose for contradiction there exists a node u with $\text{indeg}(u) > \text{outdeg}(u)$ and for every node v we have $\text{indeg}(v) \geq \text{outdeg}(v)$. Then $\sum_v \text{indeg}(v) > \sum_v \text{outdeg}(v)$ since every term on the left side is greater than or equal to the corresponding term on the right side, and at least one term (namely for $v = u$) is strictly greater. This contradicts the fact that $\sum_v \text{indeg}(v) = \sum_v \text{outdeg}(v)$ since both sides count the number of edges in the graph.

The converse also holds, by simply interchanging the roles of indegree and outdegree.

Exercise B.21.b: Consider any binary matrix, say of size $m \times n$, and suppose for contradiction that $\geq n/2$ columns each have $\geq m/2$ many 1s, and every row has $< n/4$ many 1s. Looking at the columns tells us the total number of 1s in the matrix is $\geq (n/2) \cdot (m/2) = mn/4$. Looking at the rows tells us the total number of 1s in the matrix is $< m \cdot (n/4) = mn/4$. This is a contradiction, since the number of 1s can't be both $\geq mn/4$ and $< mn/4$.

The converse does not hold: $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$ is a counterexample, in which at least one row is at least a quarter 1s, but fewer than half of the columns are at least half 1s.

Exercise B.22: This is true. Let's rephrase the "or" as an implication: For every graph $G = (V, E)$, if G is not connected then $\overline{G}$ is connected. We give a direct proof. Suppose G is not connected, so for some nodes s and t, there is no walk between s and t. Let S be the set of nodes reachable from s in G (that is, $S = \{v \in V : \exists \text{ walk between } s \text{ and } v\}$), and let $T = V \setminus S$. Then $s \in S$ and $t \in T$. For all $x \in S$ and $y \in T$, $\{x, y\}$ is not an edge in G (since otherwise y would be reachable from s by a walk to x followed by the edge $\{x, y\}$), and hence $\{x, y\}$ is an edge in $\overline{G}$.

To see that $\overline{G}$ is connected, consider any two nodes u and v. If $u \in S$ and $v \in T$, or if $u \in T$ and $v \in S$, then $u — v$ is a walk (of length 1) in $\overline{G}$. If $u, v \in S$ then $u — t — v$ is a walk (of length 2) in $\overline{G}$. If $u, v \in T$ then $u — s — v$ is a walk (of length 2) in $\overline{G}$.

Exercise B.23: Consider any partition of $[n]$ into $S_1, \ldots, S_k$. Let i be the index of a smallest set in the partition, so $|S_i| = \min(|S_1|, \ldots, |S_k|)$. First, we claim that $|S_i| \leq n/k$: This is because if $|S_i| > n/k$ then we would have the contradiction

$$n = |[n]| = |S_1| + \cdots + |S_k| \geq |S_i| + \cdots + |S_i| = k \cdot |S_i| > k \cdot n/k = n$$

where the second inequality is by the addition principle. Now, consider any $T \subseteq [n]$ with $|T| > n/k$. Since $|T| > |S_i|$ we must have $T \not\subseteq S_i$, so there exists $x \in T \setminus S_i$. Since $x \in T \subseteq [n] = S_1 \cup \cdots \cup S_k$ and $x \notin S_i$, we must have $x \in S_j$ for some $j \neq i$. Since $x \in T \cap S_j$, we have $T \cap S_j \neq \emptyset$.

Exercise B.24.a: The following algorithm finds a path of length d.

let u_1 be an arbitrary node
for $i \leftarrow 1, 2, \ldots, d$:
    pick an arbitrary neighbor u_{i+1} of u_i, with $u_{i+1} \notin \{u_1, \ldots, u_{i-1}\}$
output the path $(u_1 - u_2 - \cdots - u_d - u_{d+1})$

This maintains the invariant that $u_1 - u_2 - \cdots - u_i$ is a path. The only thing to verify is that there exists a suitable u_{i+1} inside the loop. Since $i \leq d$ inside each iteration, and since u_i has at least d neighbors, indeed at least one of u_i's neighbors must not be among the (at most $d-1$) nodes $\{u_1, \ldots, u_{i-1}\}$. Since the invariant holds at termination (with $i = d+1$), the output is a path of length d.

Exercise B.24.b: Pick an arbitrary node u. Since $\deg(u) \geq d$, u has at least d neighbors $v_1, v_2, \ldots, v_d$. For each i, since $\deg(v_i) \geq d$, v_i has at least $d-1$ neighbors $w_{i,1}, w_{i,2}, \ldots, w_{i,d-1}$ besides u. If $w_{i,k} = v_j$ for some $i, j \in [d]$ and $k \in [d-1]$, then there is a cycle $u - v_i - v_j - u$ of length 3. Otherwise, if $w_{i,k} = w_{j,\ell}$ for some distinct $i, j \in [d]$ and some $k, \ell \in [d-1]$, then there is a cycle $u - v_i - w_{i,k} - v_j - u$ of length 4. Otherwise,

$$u \text{ and } v_1, \ldots, v_d \text{ and } w_{1,1}, \ldots, w_{1,d-1} \text{ and } \cdots \text{ and } w_{d,1}, \ldots, w_{d,d-1}$$

are all distinct nodes, of which there are $1 + d + d \cdot (d-1) = d^2 + 1$.

Exercise B.25: Consider any discrete probability space, with finite sample space S.

$\Leftarrow$: We give a contrapositive proof. Suppose not every outcome has probability $\leq 2/3$. That is, $\mathbf{Pr}[s] > 2/3$ for some $s \in S$. We show that there does not exist an event whose probability is $\leq 2/3$ and $\geq 1/3$, in other words, for every event $A \subseteq S$ either $\mathbf{Pr}[A] > 2/3$ or $\mathbf{Pr}[A] < 1/3$. Let A be an arbitrary event, and consider two cases: If $s \in A$ then $\mathbf{Pr}[A] \geq \mathbf{Pr}[s] > 2/3$. If $s \notin A$ then $\mathbf{Pr}[\overline{A}] \geq \mathbf{Pr}[s] > 2/3$ and thus $\mathbf{Pr}[A] = 1 - \mathbf{Pr}[\overline{A}] < 1/3$.

$\Rightarrow$: We give a direct proof. Assume $\mathbf{Pr}[s] \leq 2/3$ for every $s \in S$. Consider two cases: If $\mathbf{Pr}[s] \geq 1/3$ for some $s \in S$, then $1/3 \leq \mathbf{Pr}[A] \leq 2/3$ holds for the event $A = \{s\}$. For the other case, suppose $\mathbf{Pr}[s] < 1/3$ for every $s \in S$. This algorithm finds an event $A \subseteq S$ such that $1/3 \leq \mathbf{Pr}[A] \leq 2/3$:

```
initialize A ← ∅
while Pr[A] < 1/3:
    pick an arbitrary s ∈ S ∖ A and add s to A  (that is, A ← A ∪ {s})
output A
```

The invariant is that $\mathbf{Pr}[A] \leq 2/3$, which trivially holds at the beginning since $\mathbf{Pr}[\emptyset] = 0$. To see that the invariant is maintained, suppose it holds right before some iteration. At this moment, if $\mathbf{Pr}[A] \geq 1/3$ then the loop would terminate without another iteration. Otherwise, $\mathbf{Pr}[A] < 1/3$ and since $\mathbf{Pr}[s] < 1/3$ by assumption, we have $\mathbf{Pr}[A \cup \{s\}] = \mathbf{Pr}[A] + \mathbf{Pr}[s] < 1/3 + 1/3 = 2/3$, so the new A at the end of the iteration indeed has $\mathbf{Pr}[A] \leq 2/3$. We should also verify that there exists $s \in S \setminus A$ for the algorithm to pick during this iteration: If not then we would have $A = S$, which would yield a contradiction since $\mathbf{Pr}[S] = 1$ but $\mathbf{Pr}[A] < 1/3$. The loop terminates because S is finite and $|A|$ grows in each iteration (so there are at most $|S|$ many iterations). When the loop terminates, $\mathbf{Pr}[A] \leq 2/3$ by the invariant, and $\mathbf{Pr}[A] \geq 1/3$ by the termination condition.

We assumed the sample space was finite. If it's countably infinite (for example, $S = \mathbb{N}$), then the proof has a flaw: The loop might not terminate. For example, infinitely many outcomes might have probability 0, and if the algorithm always picks those to add to A, then $\mathbf{Pr}[A]$ would never become $\geq 1/3$. To fix the proof, we just need to ensure that every $s \in S$ eventually gets a turn to be added to A. For example, if $S = \mathbb{N}$ then the algorithm could consider all the outcomes $0, 1, 2, \ldots$ in order, so $\mathbf{Pr}[A]$ would converge to 1 (and thus eventually become $\geq 1/3$) as more and more outcomes are added to it.

Exercise B.26: For intuition, assume that for each game, the loser is the team that came from the left subtree. Thus the rightmost leaf is the overall winner, and for each game the losing team corresponds to the rightmost leaf in the left subtree.

The formal proof is: Let S be the set of all internal nodes and T be the set of all leaves except the rightmost. Define $f : S \to T$ by $f(v) =$ the leaf reached by walking to the left child of v, then from there repeatedly walking to right children. We claim f is a bijection, which implies that $|S| = |T|$. Consider any leaf $w \in T$. We need to show that w has exactly one preimage under f. Consider the unique path from the root to w. Since w is not the rightmost leaf, the path must go to a left child at some step. Let v be the deepest node on the path such that the path goes to the left child of v. Then $f(v) = w$, because after going to v's left child, the path can only go to right children and must end at w, so w is indeed the rightmost leaf in v's left subtree. To show that v is the only preimage of w, we consider any internal node $u \in S$ such that $u \neq v$, and show that $f(u) \neq w$. If u is not on the path from the root to w, then $f(u) \neq w$ since w isn't in u's subtree. If u is below v on the path, then $f(u) \neq w$ since w is in u's right subtree. If u is above v on the path, then $f(u) \neq w$ since the path from u to w steps left upon reaching v, which means w can't be the rightmost leaf in u's left subtree.

Exercise B.27.a: The following algorithm finds a path that visits all nodes.

pick an arbitrary node u and initialize P as the path (u)
while P does not visit every node:
    pick an arbitrary node v that's not on P
    suppose $P = (u_1 \to u_2 \to \cdots \to u_k)$
    if (v, u_1) is an edge: update P to be $(v \to u_1 \to \cdots \to u_k)$
    else if (u_k, v) is an edge: update P to be $(u_1 \to \cdots \to u_k \to v)$
    else:
        find an index $i \in [k-1]$ such that both (u_i, v) and (v, u_{i+1}) are edges
        update P to be $(u_1 \to \cdots \to u_i \to v \to u_{i+1} \to \cdots \to u_k)$
output P

This maintains the invariant that P is a path. The invariant holds before the first iteration, so it holds after every iteration. The loop terminates since the length of P increases in every iteration but can't exceed the number of nodes minus 1. When the loop terminates, P is a path (by the invariant) that visits every node (by the termination condition). The only thing to justify is that a suitable index i in the "else" case is guaranteed to exist. The following mini-algorithm finds such an i, assuming (v, u_1) and (u_k, v) are not edges, which means (u_1, v) and (v, u_k) are edges:

initialize $i = 1$
while (v, u_{i+1}) is not an edge: increment i
output i

This maintains the invariant that (u_i, v) is an edge (because in any iteration, if (v, u_{i+1}) is not an edge then (u_{i+1}, v) is an edge). The invariant holds for $i = 1$ by assumption, so it must hold for every i until the loop terminates. The loop must terminate when $i = k - 1$ or before, since (v, u_k) is an edge by assumption. Thus for the output i, (u_i, v) is an edge (by the invariant) and (v, u_{i+1}) is an edge (by the termination condition).

Exercise B.27.b: By definition, a round-robin tournament can't have a cycle of length < 3. Supposing a round-robin tournament has a cycle, the following algorithm finds a cycle of length 3.

initialize C as a cycle in the graph
while C has length > 3:
    suppose $C = (u_1 \to u_2 \to u_3 \to \cdots \to u_k \to u_1)$
    if (u_3, u_1) is an edge: update C to be $(u_1 \to u_2 \to u_3 \to u_1)$
    if (u_1, u_3) is an edge: update C to be $(u_1 \to u_3 \to \cdots \to u_k \to u_1)$ (that is, bypassing u_2)
output C

This maintains the invariant that C is a cycle. The invariant holds before the first iteration, so it holds after every iteration. The loop terminates since one of the two "if" conditions inside each loop iteration must hold (because the graph is a round-robin tournament) and so the length of C decreases in every iteration. When the loop terminates, C is a cycle (by the invariant) of length 3 (by the termination condition).

Exercise B.27.c: Let u be a node with highest outdegree, and consider an arbitrary node v. If $v = u$ then there is a path of length 0 from u to v. Let S be the set of nodes that have an edge from u. If $v \in S$ then there is a path of length 1 from u to v. Otherwise, suppose $v \neq u$ and $v \notin S$. Then we claim there is an edge from some node in S to v, which implies there is a path of length 2 from u to v. Suppose for contradiction that no node in S has an edge to v. Since the graph is a round-robin tournament, v must have edges to every node of S and to u. We conclude that $\text{outdeg}(v) \geq |S| + 1 = \text{outdeg}(u) + 1$, contradicting the assumption that u has highest outdegree among all nodes.

Exercise B.28: Consider any undirected graph $G = (V, E)$, which may have multi-edges and self-loops.

$\Rightarrow$: Suppose there exists a partition of G's edges into cycles. Consider any cut S, T (that is, $V = S \cup T$ and $S \cap T = \emptyset$). For each cycle in the partition, start at an arbitrary node in the cycle and traverse the cycle in an arbitrary direction until returning to the starting node. This walk may alternate between S and T, but since it ends up on the same side of the cut as it started, it must alternate an even number of times. That is, an even number of the cycle's edges cross the cut. Since each cycle in the partition contributes an even number of edges crossing the cut, and each edge crossing the cut is accounted for exactly once (since the cycles partition E), the number of edges crossing the cut is a sum of even numbers and is hence even.

$\Leftarrow$: Suppose every cut is crossed by an even number of G's edges. This algorithm finds a partition of E into cycles:

repeat until no edges remain:
  pick any edge $\{v_1, v_2\}$ and walk arbitrarily, without reusing edges, until revisiting a node
  say the walk is $(v_1 — \cdots — \underbrace{v_k — \cdots — v_\ell = v_k}_{\text{cycle}})$
  output the cycle $(v_k — \cdots — v_\ell = v_k)$ and remove these edges from G

As long as "walk arbitrarily, without resuing edges" never gets stuck, it always finds a cycle. Since the cycle's edges are removed from G, every edge of G appears in at most one of the cycles output by the algorithm. Since the algorithm only terminates when no edges remain, every edge of G appears in at least one of the cycles output by the algorithm. Thus the algorithm indeed finds a partition of E into cycles.

Not getting stuck means there is always an unused edge incident to the current node, before any node is revisited. The only way to get stuck is by reaching a node of degree 1, since leaving the node would require reusing the edge along which the node was entered. But if $\deg(v) = 1$ then $S = \{v\}$, $T = V \setminus \{v\}$ would be a cut crossed by an odd number of edges (namely, one edge).

Thus as long as every cut is crossed by an even number of remaining edges, the algorithm won't get stuck. So it suffices to argue that the algorithm maintains the (outer) loop invariant that every cut is crossed by an even number of remaining edges. This holds at the beginning by assumption. Each iteration maintains the invariant because every cut is crossed by an even number of edges of the cycle being removed (by the $\Leftarrow$ direction) and thus the number of edges crossing the cut after the cycle's removal is a difference of two even numbers and is hence even.

Exercise B.29: $\Rightarrow$: Suppose G has a full trail from s to t. To see that G is weakly connected except for isolated nodes, consider any two nonisolated nodes u and v. The full trail eventually visits both u and v because it traverses the edge(s) touching u and the edge(s) touching v. Thus there exists a trail either from u to v or from v to u (namely, the segment of the full trail between the first visit to u or v, and the first visit to the other). Ignoring the arrow directions, this gives a trail between u and v in the undirected version of G.

To see that $\text{indeg}(v) = \text{outdeg}(v)$ for each node $v \notin \{s, t\}$, note that each time the full trail transits through v, it uses one incoming edge and one outgoing edge. Thus v's incoming edges are paired up with its outgoing edges, so there must be the same number of each.

To see that $\text{outdeg}(s) = \text{indeg}(s) + 1$, note that the full trail begins with an outgoing edge of s, and each subsequent transit through s uses one incoming edge and one outgoing edge. Thus s's incoming edges are paired up with all but one of its outgoing edges.

To see that $\text{indeg}(t) = \text{outdeg}(t) + 1$, note that the full trail ends with an incoming edge of t, and each prior transit through t uses one incoming edge and one outgoing edge. Thus t's outgoing edges are paired up with all but one of its incoming edges.

$\Leftarrow$: The first phase loop maintains the invariant that the sequence of edges followed forms a trail from s. Each iteration can't get stuck:

- If the current node v is not s (and not t, since otherwise the loop would have terminated), then there indeed exists an unused outgoing edge from v because $\text{outdeg}(v) = \text{indeg}(v)$ and prior transits through v used the same number of incoming and outgoing edges, and the current entry into v used one more incoming edge.
- If the current node is s being revisited, then there indeed exists an unused outgoing edge from s because $\text{outdeg}(s) = \text{indeg}(s) + 1$ and the number of outgoing used edges equals the number of incoming used edges: Prior transits through s used the same number of incoming and outgoing edges, and the current entry into s used one more incoming edge, and the very first iteration used one more outgoing edge.
- If the current node is s in the first iteration, then there indeed exists an unused outgoing edge from s because $\text{outdeg}(s) = \text{indeg}(s) + 1 \geq 1$.

The loop terminates since the trail length grows in each iteration but can't exceed the total number of edges in G (since each edge is used at most once). When this loop terminates, the last node is t.

The second phase's outer loop maintains the invariant that it has a sequence of edges that forms an s-t trail. We just argued that this invariant holds at the beginning of the second phase. This invariant implies that every node, including s and t, has the same number of unused incoming edges as unused outgoing edges:

- For $v \notin \{s, t\}$, we have $\text{indeg}(v) = \text{outdeg}(v)$ and all transits through v on the current s-t trail pair up incoming with outgoing edges, which means v has the same number of used incoming edges as used outgoing edges.
- $\text{outdeg}(s) = \text{indeg}(s) + 1$ and all transits through s pair up incoming with outgoing edges, and the trail's first edge is outgoing from s, which means s has one more used outgoing edge than used incoming edges.

- $\text{indeg}(t) = \text{outdeg}(t) + 1$ and all transits through t pair up incoming with outgoing edges, and the trail's last edge is incoming to t, which means t has one more used incoming edge than used outgoing edges.

To see that the invariant is maintained, assume it holds at the beginning of some outer iteration.

First, we claim that there indeed exists a node u that's visited by the current s-t trail and has an unused outgoing edge. There exists some unused edge in G (otherwise the second phase would have terminated), and since G is weakly connected, there must be an undirected path from s that ends with the undirected counterpart of this unused edge. Following this path from s, let $\{u, v\}$ be the first undirected edge whose directed counterpart is unused (not traversed by the current s-t trail). At least one of the nodes u or v must be on the current s-t trail; assume it's u. If the directed counterpart is (u, v) then we're done proving the claim because (u, v) is an unused outgoing edge from u. If the directed counterpart is (v, u) (which is an unused incoming edge to u) then there must also exist some unused outgoing edge from u, because u has the same number of unused incoming edges as unused outgoing edges (due to the outer loop invariant, as we argued earlier).

Now that we know the algorithm can find such a u, we argue that the second phase's inner loop maintains the invariant that the sequence of edges followed forms a trail from u consisting of edges unused by the current s-t trail. Each iteration can't get stuck: For the first inner iteration, we already know u has an unused outgoing edge. For any other inner iteration, suppose the current node is some v (unrelated to the previous paragraph). Then v has the same number of incoming and outgoing edges unused by the current s-t trail, and the same number of incoming and outgoing edges used by prior transits through v on the current trail from u, and one more incoming edge used as the latest entry to v. Thus v must have an outgoing edge that's unused (by both the current s-t trail and the current trail from u).

The inner loop terminates (by reaching u again) because the trail length grows in each iteration but can't grow forever. The splicing operation successfully maintains the outer invariant since the s-t trail and the closed trail have node u in common and no edges in common. The second phase's outer loop terminates since the s-t trail length grows in each iteration (by at least two edges touching u) but can't grow forever. Since every edge is used at termination and the outer invariant holds at termination, the output is indeed a full s-t trail.

Exercise B.30.a: We rephrase the proposition by "rewinding" the deletion: Adding an edge can make the number of connected components go down by at most one. For each node w, let V_w denote w's connected component (the set of nodes reachable from w) before we add edge $e = \{u, v\}$ to the graph.

If u and v already had the same connected component ($V_u = V_v$), then none of the connected components in the graph change after adding e. This is because if two nodes are reachable from each other after adding e, then they were reachable from each other before adding e, by rerouting (if necessary) along a walk between u and v that doesn't use e.

On the other hand, suppose u and v had different connected components ($V_u \cap V_v = \emptyset$). After adding e, V_u and V_v are merged into one connected component containing both u and v. This is because in addition to reaching all of V_u, u can now also reach all of V_v by first taking the edge to v; thus u's (and v's) new connected component contains all of $V_u \cup V_v$, and it contains nothing more because any node reachable from u with the help of e must have been reachable from either u or v already. Furthermore, none of the other connected components change: If $w \notin V_u \cup V_v$, then w's connected component remains V_w (since w can reach everything it could before the addition, and it can't reach anything new with the help of e since otherwise u or v would already have been reachable from w). Since the only effect is that two connected components are merged into one, the number of connected components goes down by only one.

Exercise B.30.b: Suppose for contradiction there exists a connected graph with n nodes and at most $n-2$ edges. After deleting all the edges (but none of the nodes) one by one, the number of connected components goes from 1 (since the original graph is connected) to at most $1+(n-2) = n-1$ since the number goes up by at most 1 after each edge deletion (Exercise B.30.a). On the other hand, after deleting all the edges, the number of connected components is n since each node becomes its own connected component. This is a contradiction: The number of connected components can't be both n and $\leq n-1$.

Exercise B.30.c: We give a contrapositive proof of this rephrasing: For every connected graph with at least two nodes, if it's a tree then it has at least one node with degree 1.

Suppose every node has degree $\neq 1$. Since the graph is connected and has at least two nodes, no node can have degree 0. Thus every node has degree ≥ 2. The following algorithm finds a cycle, thus showing that the graph is not a tree. We assume that multi-edges are not present, since otherwise there would trivially be a cycle of length 2.

let $\{u_1, u_2\}$ be an arbitrary edge
for $i \leftarrow 2, 3, 4, \ldots$:
    pick an arbitrary edge incident to u_i except $\{u_{i-1}, u_i\}$, and let u_{i+1} be the other endpoint
    if $u_{i+1} = u_j$ for some $j \leq i$: halt and output the cycle $(u_j — u_{j+1} — \cdots — u_i — u_j)$

This maintains the invariant that $u_1 — u_2 — \cdots — u_i$ is a path, since it terminates as soon as it revisits a node. In each iteration, there indeed exists an edge incident to u_i other than $\{u_{i-1}, u_i\}$, since $\deg(u_i) \geq 2$ (so the algorithm doesn't get "stuck"). The invariant trivially holds before the first iteration, so it holds after every iteration. The loop terminates since there are only finitely many nodes (so revisiting is inevitable). When the loop terminates during the last iteration, the output is a cycle since $u_j — u_{j+1} — \cdots — u_i$ is a path (by the invariant) and $u_i — u_j$ is an edge (by the termination condition).

Exercise B.30.d: Consider an arbitrary connected graph G with n nodes.

$\Leftarrow$: We give a contrapositive proof. Suppose G is not a tree. By Theorem B.21, there exists an edge e whose deletion would not disconnect G. Since G with e deleted is connected, it has at least $n-1$ edges (Exercise B.30.b). Also counting e itself, G must have at least n edges.

$\Rightarrow$ First proof: We give a direct proof. Suppose G is a tree. Consider an algorithm that repeatedly finds a node with degree 1 and deletes it (along with its incident edge), until only one node remains. Letting H be the variable representing the intermediate graph throughout the algorithm, the algorithm maintains the invariant that H is a tree and

$$(\#\text{ nodes of } H) - (\#\text{ edges of } H) \;=\; (\#\text{ nodes of } G) - (\#\text{ edges of } G)$$

which holds at the beginning when $H = G$. To see that the invariant is maintained, suppose it holds at the beginning of some iteration. Since H is a tree with at least two nodes, it indeed has a node with degree 1 (Exercise B.30.c), so the algorithm doesn't get stuck. After deleting this node and edge, the remaining graph is still a tree since it's connected (no path could use the deleted edge unless it ends at the deleted node) and no cycles were introduced. Furthermore, the number of nodes and number of edges of H both went down by 1, so their difference remains unchanged. So the invariant holds after this iteration. The loop terminates when H has one node and zero edges, so the invariant tells us $1 - 0 = (\#\text{ nodes of } G) - (\#\text{ edges of } G)$. Thus $(\#\text{ edges of } G) = n - 1$.

$\Rightarrow$ Second proof: We give a direct proof. Suppose G is a tree. Consider an algorithm that deletes all the edges (but none of the nodes) one by one, in an arbitrary order. This maintains the invariant that every connected component in the intermediate graph is a tree, since no cycles are ever introduced. Thus by Theorem B.21, each edge deletion disconnects one connected component. So the number of connected components starts out as 1 and goes up by at least 1 (in fact, exactly 1 by Exercise B.30.a) in each iteration, and ends at n when no edges remain (each node is its own connected component). Going from 1 to n, by adding 1 at a time, only takes $n-1$ steps, so there must have been only $n-1$ edges to delete.

Exercise B.31: $\geq$: For every topological layering and every path, the nodes on the path must all be in different layers (the first node on the path is in a lower-index layer than the second, which is in a lower index layer than the third, and so on). Thus the number of layers is at least the number of nodes on the path, which is one plus the length of the path.

$\leq$: Define k as the length of a longest path in the dag G. The following algorithm finds a topological layering with at most $k+1$ layers. To aid the analysis, the algorithm explicitly names the intermediate graphs $H_1, H_2, \ldots$ rather than using a single variable H.

```
let H_1 = G
for i = 1, 2, ... until H_i is empty:
    let L_i be the set of all sources of H_i
    obtain H_{i+1} from H_i by deleting all nodes of L_i (and their outgoing edges)
```

This maintains the invariant that H_i is a dag and that every path in H_i has length $\leq k-i+1$, which holds at the beginning when $i = 1$. Suppose the invariant holds for H_i. Then H_{i+1} is certainly a dag since H_i is. Furthermore, we claim that H_{i+1} can't have a path of length $> k-(i+1)+1$. Suppose for contradiction H_{i+1} has a path of length $k-i+1$, and let v be the starting node of the path. Then v can't have been a source in H_i, since otherwise $v \in L_i$ so v wouldn't have been included in H_{i+1}. Thus v has an incoming edge in H_i, and prepending this edge to the path produces a path of length $> k-i+1$ in H_i, contradicting the assumption that every path in H_i has length $\leq k-i+1$. So the invariant also holds for H_{i+1}.

The loop terminates since Lemma B.22 tells us there indeed always exists a source in H_i (so the algorithm must eventually run out of nodes to delete). Let ℓ be the largest value of i for which H_i is nonempty (right before the loop terminates). Then we know the nodes of G are partitioned into layers $L_1, L_2, \ldots, L_\ell$, and since every path in H_ℓ has length $\leq k-\ell+1$ (by the invariant), we must have $k-\ell+1 \geq 0$, so the number of layers is $\ell \leq k+1$.

To see that $L_1, L_2, \ldots, L_\ell$ is a topological layering of G, consider any edge (u, v) of G, where $u \in L_i$ and $v \in L_j$. We need to show that $i < j$, so suppose for contradiction that $j \leq i$. Then u and v are both in H_j (since u hasn't been deleted yet), so v has an incoming edge (from u) and is thus not a source in H_j, contradicting the assumption that $v \in L_j$.

Exercise B.32: Consider any such p and S. Pick a uniformly random $x \in [p-1]$, and view $(x, -x, x^{-1})$ as an outcome of a joint distribution over $[p-1] \times [p-1] \times [p-1]$. The marginal distribution of $-x$ is uniform since each element of $[p-1]$ is the additive inverse of exactly one element of $[p-1]$. The marginal distribution of x^{-1} is also uniform, for a similar reason. Thus $\mathbf{Pr}[x \notin S] = \mathbf{Pr}[-x \notin S] = \mathbf{Pr}[x^{-1} \notin S] = 1 - |S|/(p-1) < 1 - 2/3 = 1/3$. By a union bound, $\mathbf{Pr}[x \notin S \text{ or } -x \notin S \text{ or } x^{-1} \notin S] < 1/3 + 1/3 + 1/3 = 1$, so $\mathbf{Pr}[x \in S \text{ and } -x \in S \text{ and } x^{-1} \in S] > 0$. Thus there exists x such that $x, -x, x^{-1} \in S$.

Exercise B.33: Consider any such S. For $x \in \{0,1\}^n$ and $i \in [n]$, let's have $x^i \in \{0,1\}^n$ denote the neighbor of x with the i^{th} bit flipped. Pick a uniformly random x, and view $(x, x^1, \ldots, x^n)$ as an outcome of a joint distribution over $(\{0,1\}^n)^{n+1}$. For each i, the marginal distribution of x^i is uniform since the i^{th} bit remains uniformly random and independent of the other bits. Thus $\mathbf{Pr}[x \notin S] = 1 - |S|/2^n < \frac{1}{n+1}$ and similarly $\mathbf{Pr}[x^i \notin S] < \frac{1}{n+1}$ for each i. By a union bound, $\mathbf{Pr}[x \notin S \text{ or } x^1 \notin S \text{ or } \cdots \text{ or } x^n \notin S] < \frac{1}{n+1} + \cdots + \frac{1}{n+1} = 1$. Thus there exists x such that $x \in S$ and $x^1 \in S$ and $\cdots$ and $x^n \in S$.

Exercise B.34: Let $k = \lceil 2 \log n \rceil + 1$. We may assume n is large enough that $k \leq n$, since otherwise the theorem is trivially true. Pick a uniformly random graph G with n nodes. For each $S \subseteq [n]$ with $|S| = k$, let A_S be the event that S is an independent set or clique in G. Then $\mathbf{Pr}[A_S] = 2 \cdot 2^{-k(k-1)/2}$ because A_S can be partitioned into two events corresponding to S being an independent set or being a clique, and both of these events have probability $(1/2)^{k(k-1)/2}$ (since the $\binom{k}{2} = k(k-1)/2$ many pairs of distinct nodes in S each has probability $1/2$ of being an edge, independently for all pairs). There are $\binom{n}{k}$ many events. We have $\binom{n}{k} < n^k/2$ as in the proof of Theorem B.25. The definition of k implies $k \geq 2 \log n + 1$, which rearranges to $n \leq 2^{(k-1)/2}$. Putting everything together with a union bound:

$$\mathbf{Pr}\left[\bigcup_S A_S\right] \leq \sum_S \mathbf{Pr}[A_S] = \binom{n}{k} \cdot 2 \cdot 2^{-k(k-1)/2} < n^k \cdot 2^{-k(k-1)/2} \leq (2^{(k-1)/2})^k \cdot 2^{-k(k-1)/2} = 1$$

Thus there exists an outcome G that's in none of the A_S events, so G has no independent set or clique of size exactly k, which also implies G has no independent set or clique of size $\geq k$.

Exercise B.35.a: We give a direct proof. Assuming every row of M has at least $n/2$ many 1s, the following algorithm finds a desired set of columns. It maintains the invariant that J is a set of column indices, and I is the set of indices of all rows that have no 1 in any of the J columns (that is, rows that have yet to be "taken care of"), and furthermore, after the h^{th} iteration, $|I| \leq n/2^h$ and $|J| = h$.

initialize $I \leftarrow [n]$ and $J \leftarrow \emptyset$
while $I \neq \emptyset$:
    find a column that's at least half 1s in the submatrix $M_{I,[n]\setminus J}$
    let j be the index of that column in M
    update J to include j, and I to exclude every i such that $M_{i,j} = 1$
output J

First, we claim that there always exists a column that's at least half 1s in the submatrix $M_{I,[n]\setminus J}$. Since each row $i \in I$ has at least $n/2$ many 1s in M but has no 1s among the J columns (by the invariant), the row has at least $n/2 \geq (n - |J|)/2$ many 1s among the $[n] \setminus J$ columns. That is, each row of $M_{I,[n]\setminus J}$ is at least half 1s. By Theorem B.17, $M_{I,[n]\setminus J}$ has a column that's at least half 1s.

In each iteration, since column j is at least half 1s among the I rows, and rows with 1 in column j get excluded from I, this means $|I|$ shrinks by at least a factor of 2. So if $|I| \leq n/2^{h-1}$ before the h^{th} iteration, then $|I| \leq \frac{1}{2} \cdot n/2^{h-1} = n/2^h$ after the h^{th} iteration, and thus the invariant is maintained. Letting $k = \lfloor \log n \rfloor + 1 > \log n$, we would have $|I| < n/2^{\log n} = 1$ after the k^{th} iteration, and thus the loop terminates with $|J| \leq k$. Since $I = \emptyset$ at termination, the invariant tells us there is no row that has no 1 in any of the J columns—that is, every row has a 1 in at least one J column.

Exercise B.35.b: Assume at least half of M's rows each have at least $n/2$ many 1s. (Otherwise, at least half of M's rows would each have at least $n/2$ many 0s, in which case the same argument would work, with the roles of 0 and 1 interchanged.) The following algorithm finds a submatrix with at least k rows and k columns, such that every entry is 1. It maintains the invariant that every entry of $M_{I,J}$ is 1 and that after the h^{th} iteration, $|I| \geq (n/2)/3^h$ and $|J| = h$.

initialize $I \leftarrow \{i : \text{the number of 1s in row } i \text{ is } \geq n/2\}$ and $J \leftarrow \emptyset$
while $|J| < k$:
    find a column that's at least a third 1s in the submatrix $M_{I,[n]\setminus J}$
    let j be the index of that column in M
    update J to include j, and I to exclude every i such that $M_{i,j} = 0$
output I, J

First, we claim that there always exists a column that's at least a third 1s in the submatrix $M_{I,[n]\setminus J}$. Since each row $i \in I$ has at least $n/2$ many 1s in M, the row has at least $n/2 - |J|$ many 1s among the $[n] \setminus J$ columns. We have $n/2 - |J| \geq n/2 - k \geq (n - \frac{1}{2}\log n)/2 \geq n/3$. Thus, the fraction of 1s in each row of $M_{I,[n]\setminus J}$ is $\geq (n/3)/(n - |J|) \geq (n/3)/n = 1/3$. That is, each row of $M_{I,[n]\setminus J}$ is at least a third 1s. By a simple variant of Theorem B.17, $M_{I,[n]\setminus J}$ has a column that's at least a third 1s.

In each iteration, since column j is at least a third 1s among the I rows, and rows with 0 in column j get excluded from I, this means $|I|$ shrinks by at most a factor of 3. So if $|I| \geq (n/2)/3^{h-1}$ before the h^{th} iteration, then $|I| \geq \frac{1}{3}(n/2)/3^{h-1} = (n/2)/3^h$ after the h^{th} iteration, and thus the invariant is maintained. Since the invariant holds before the first iteration ($|I| \geq n/2 = (n/2)/3^0$ at the beginning), it holds after all iterations. With $k = \lfloor \frac{1}{2}\log n \rfloor \leq \frac{1}{2}\log n$, after the k^{th} iteration we have

$$|I| \;\geq\; \tfrac{n}{2}/3^{\frac{1}{2}\log n} \;=\; \tfrac{n}{2}/2^{(\log 3)\cdot\frac{1}{2}\log n} \;=\; \tfrac{n}{2}/n^{(\log 3)\cdot\frac{1}{2}} \;\geq\; \tfrac{n}{2}/n^{4/5} \;=\; \tfrac{1}{2}n^{1/5} \;\geq\; \tfrac{1}{2}\log n \;\geq\; k$$

assuming n is sufficiently large. Thus the loop terminates with $|I| \geq k$ and $|J| = k$, and every entry of $M_{I,J}$ is 1. To get a submatrix of size exactly $k \times k$, any $|I| - k$ many rows can be deleted from I.

Exercise B.36.a: Consider any such x. Pick a uniformly random $k \in \mathbb{Z}_n$, and define y by rotating x right by k positions, so $y_i = x_{(i-k) \bmod n}$. View (k, y) as an outcome of a joint distribution over $\mathbb{Z}_n \times \{0,1\}^n$. For each $i \in [n]$, since $(i-k) \bmod n$ is uniformly distributed over $\mathbb{Z}_n$, and there are $< \sqrt{n}$ many values of k with $x_{(i-k) \bmod n} = 1$, we have $\mathbf{Pr}[y_i = 1] < \sqrt{n}/n = 1/\sqrt{n}$. For each i such that $x_i = 1$ (of which there are $< \sqrt{n}$), let A_i be the event that $y_i = 1$. Then the probability that $x \wedge y$ is not the all-0 string equals

$$\mathbf{Pr}[\exists i : x_i = y_i = 1] = \mathbf{Pr}\left[\bigcup_{i : x_i = 1} A_i\right] \leq \sum_{i : x_i = 1} \mathbf{Pr}[A_i] < \sum_{i : x_i = 1} 1/\sqrt{n} < \sqrt{n} \cdot 1/\sqrt{n} = 1$$

by a union bound. Thus there exists an outcome (k, y) that has positive probability (so y is a rotation of x) and $x \wedge y$ is the all-0 string.

Exercise B.36.b: Partition the index set $[n]$ into $\sqrt{n}$ many contiguous blocks, each of size $\sqrt{n}$. Define x as follows: The first block is all 1s. Each of the other blocks starts with a single 1, and the rest of the bits are 0s. For example, $x = 111100100$ if $n = 9$.

The number of 1s in this x is $\sqrt{n}$ (from the first block) plus $\sqrt{n}-1$ (from the single 1s in the other $\sqrt{n}-1$ many blocks), so indeed $\text{weight}(x) = 2\sqrt{n}-1 < 2\sqrt{n}$. There is no run of $\sqrt{n}$ many consecutive 0s in x, so no matter how we rotate it, the first block of the rotation will not be all 0s. In other words, for every rotation y of x, there exists $i \in [\sqrt{n}]$ such that $y_i = 1$ and of course $x_i = 1$, so $x \wedge y$ is not the all-0 string since $x_i \wedge y_i = 1$.

Exercise B.37: Consider any $x, y \in \{0,1\}^n$. Pick a uniformly random $k \in \mathbb{Z}_n$, and define z by rotating y right by k positions, so $z_i = y_{(i-k) \bmod n}$. View (k, z) as an outcome of a joint distribution over $\mathbb{Z}_n \times \{0,1\}^n$. For each i, since $(i-k) \bmod n$ is uniformly distributed over $\mathbb{Z}_n$, and there are weight(y) many values of k with $y_{(i-k) \bmod n} = 1$, and $n - \text{weight}(y)$ many values of k with $y_{(i-k) \bmod n} = 0$, we have:

$$\mathbf{Pr}[x_i \neq z_i] = \begin{cases} \text{weight}(y)/n & \text{if } x_i = 0 \\ 1 - \text{weight}(y)/n & \text{if } x_i = 1 \end{cases}$$

Defining the random variable $X = \text{dist}(x, z)$ as the number of indices i for which $x_i \neq z_i$, we have $X = X_0 + \cdots + X_{n-1}$ where X_i is the indicator random variable for the event $x_i \neq z_i$. By linearity of expectation:

$$\begin{aligned} \mathbf{E}[X] &= \textstyle\sum_{i=0}^{n-1} \mathbf{E}[X_i] = \sum_{i=0}^{n-1} \mathbf{Pr}[x_i \neq z_i] \\ &= (n - \text{weight}(x)) \cdot (\text{weight}(y)/n) + \text{weight}(x) \cdot (1 - \text{weight}(y)/n) \\ &= \text{weight}(x) + \text{weight}(y) - 2\,\text{weight}(x)\,\text{weight}(y)/n \end{aligned}$$

By Lemma B.26, there exists an outcome (k, z) such that $\text{dist}(x, z) = X(k, z)$ is at least as large (and the outcome has positive probability, so z is a rotation of y).

Exercise B.38.a: For each $d \in \{0,1,\ldots,n\}$, exactly $\binom{n}{d}$ many strings y have $\text{dist}(x,y) = d$ (since such a y is specified by choosing which d bit positions of x to flip). Since $\binom{n}{d} = \binom{n}{n-d}$, we have $\sum_{d=0}^{\lfloor n/2 \rfloor} \binom{n}{d} = \sum_{d=\lceil n/2 \rceil}^{n} \binom{n}{d}$, which implies that $\sum_{d=0}^{\lfloor n/2 \rfloor} \binom{n}{d}$ is exactly half of $\sum_{d=0}^{n} \binom{n}{d} = 2^n$. Thus $\mathbf{Pr}\big[\text{dist}(x,y) < n/2\big] = (\frac{1}{2} \cdot 2^n)/2^n = 1/2$.

An alternative proof involves partitioning $\{0,1\}^n$ into "bitwise negation pairs": For each $z \in \{0,1\}^n$ with $z_1 = 0$, $\{z, \overline{z}\}$ forms one part of the partition (so there are 2^{n-1} parts in total). Since $\text{dist}(x,z) = n - \text{dist}(x,\overline{z})$, exactly one of the pair $z, \overline{z}$ has distance $< n/2$ from x. Letting B be the event that $\text{dist}(x,y) < n/2$, and A_z (where $z_1 = 0$) be the event that $y \in \{z, \overline{z}\}$, we have $\mathbf{Pr}[A_z] = 2/2^n$ and $\mathbf{Pr}[B \mid A_z] = 1/2$. Thus by the law of total probability:

$$\mathbf{Pr}\big[\text{dist}(x,y) < n/2\big] = \sum_{z\,:\,z_1=0} \mathbf{Pr}[A_z] \cdot \mathbf{Pr}[B \mid A_z] = \sum_{z\,:\,z_1=0} \tfrac{1}{2^{n-1}} \cdot \tfrac{1}{2} = 2^{n-1} \cdot \tfrac{1}{2^{n-1}} \cdot \tfrac{1}{2} = 1/2$$

Exercise B.38.b: Sample $y \in \{0,1\}^n$ uniformly at random. Let the random variable X count the number of strings $x \in S$ with $\text{dist}(x,y) < n/2$. Thus $X = \sum_{x \in S} X_x$ where X_x is the indicator random variable for the event that $\text{dist}(x,y) < n/2$. By linearity of expectation and Exercise B.38.a:

$$\mathbf{E}[X] = \sum_{x \in S} \mathbf{E}[X_x] = \sum_{x \in S} \mathbf{Pr}[\text{dist}(x,y) < n/2] = \sum_{x \in S} 1/2 = |S|/2$$

By Lemma B.26, there exists an outcome y such that $X(y) \geq |S|/2$, which means $\text{dist}(x,y) < n/2$ for at least half of the strings $x \in S$.

Exercise B.39.a: For any index $i \in [n]$, define:

$$d(i) \;=\; (\#\text{ 1s among } x_1 \cdots x_i) - (\#\text{ 0s among } x_1 \cdots x_i) \;=\; 2 \cdot \text{weight}(x_1 \cdots x_i) - i$$

Note that $d(n) = 0$ since $\text{weight}(x) = n/2$, and that $\text{weight}(x_1 \cdots x_i) = i/2$ is equivalent to $d(i) = 0$. Observe that if $i > 1$ then $|d(i-1) - d(i)| = 1$. Specifically:

$$d(i-1) \;=\; \begin{cases} d(i)+1 & \text{if } x_i = 0 \\ d(i)-1 & \text{if } x_i = 1 \end{cases}$$

First, assume $x_1 = x_n = 1$. Then $d(1) = 1$ and $d(n-1) = -1$ (since $d(n) = 0$ and $x_n = 1$). Let i be the smallest index such that $d(i) \leq 0$. Then $i > 1$ since $d(1) > 0$, and $i \leq n-1$ since $d(n-1) \leq 0$. Since $0 < d(i-1) \leq d(i)+1 \leq 0+1$, the only possibility is $d(i-1) = 1$ and $d(i) = 0$. The case where $x_1 = x_n = 0$ is analogous: Letting i be the smallest index such that $d(i) \geq 0$, we have $i > 1$ since $d(1) < 0$, and $i \leq n-1$ since $d(n-1) \geq 0$, and $0 > d(i-1) \geq d(i) - 1 \geq 0 - 1$ forces $d(i-1) = -1$ and $d(i) = 0$.

Exercise B.39.b: For indices $1 \leq i \leq j \leq n$, define:

$$d(i,j) \;=\; (\#\text{ 1s among } x_i \cdots x_j) - (\#\text{ 0s among } x_i \cdots x_j)$$

Note that $d(1,n) = 0$ since weight$(x) = n/2$. Pick $k \in [n]$ that minimizes $d(1,k)$. If $k = n$ then we can let $y = x$ since $d(1,i) \geq d(1,k) = 0$ for all i, so assume $k < n$. Let $y = x_{k+1} \cdots x_n x_1 \cdots x_k$— that is, x rotated left by k positions. For all $i \in [n-k]$ we have $d(k+1, k+i) \geq 0$ since otherwise $d(1,k+i) = d(1,k) + d(k+1,k+i) < d(1,k)$, which would contradict the choice of k; thus $y_1 \cdots y_i = x_{k+1} \cdots x_{k+i}$ has at least as many 1s as 0s. Also, for all $j \in [k]$ we have $d(1,j) \geq d(1,k)$ and thus $d(k+1,n) + d(1,j) \geq d(k+1,n) + d(1,k) = d(1,n) = 0$; thus for $i = n-k+j$ we have $y_1 \cdots y_i = x_{k+1} \cdots x_n x_1 \cdots x_j$ has at least as many 1s as 0s.

Exercise B.40: Let $d = \gcd(a, b) > 0$ and $a = q_1 d$ and $b = q_2 d$. We claim that q_1 and q_2 are coprime. Suppose for contradiction they are not. Then for some integer $c > 1$ we have $q_1 = q_3 c$ and $q_2 = q_4 c$ for some integers q_3 and q_4. Thus $a = q_3(cd)$ and $b = q_4(cd)$. This means cd is a common divisor of a and b, and $cd > d$ since $c > 1$, which contradicts the fact that $d = \gcd(a, b)$.

We have $0 < q_1 < q_2$ since $0 < a < b$. By Theorem B.29, there exists $x \in \mathbb{Z}_{q_2}$ such that $q_1 x \bmod q_2 = 1$, which means $q_1 x = q_5 q_2 + 1$ for some integer q_5. Multiplying the equation by d, we have $ax = q_5 b + d$. Since $d \leq a < b$, this means $ax \bmod b = d$. Since $q_2 \leq b$, we have $x \in \mathbb{Z}_b$.

Exercise B.41.a: $a \cdot 1,\ a \cdot 2,\ \ldots,\ a \cdot (p-1)$ are each nonzero because if $a \cdot b = 0$ then $b = 1 \cdot b = a^{-1} \cdot a \cdot b = a^{-1} \cdot 0 = 0$. Also, $a \cdot 1,\ a \cdot 2,\ \ldots,\ a \cdot (p-1)$ are all distinct because if $a \cdot b = a \cdot c$ then $b = 1 \cdot b = a^{-1} \cdot a \cdot b = a^{-1} \cdot a \cdot c = 1 \cdot c = c$. Viewing the nonzero elements of $\mathbb{Z}_p$ as pigeons and as holes, and having pigeon b fly into hole $a \cdot b$, we just showed that no hole gets more than one pigeon. Since the number of holes equals the number of pigeons, the pigeonhole principle tells us that each hole gets exactly one pigeon. In other words, $a \cdot 1,\ a \cdot 2,\ \ldots,\ a \cdot (p-1)$ is a permutation of the numbers $1, 2, \ldots, p-1$. Thus, $a^{p-1} \cdot 1 \cdot 2 \cdots (p-1) = (a \cdot 1) \cdot (a \cdot 2) \cdots (a \cdot (p-1)) = 1 \cdot 2 \cdots (p-1)$. Multiplying both sides by $1^{-1} \cdot 2^{-1} \cdots (p-1)^{-1}$ produces $a^{p-1} = 1$.

Exercise B.41.b: First, we claim that $a^p = a$ holds for every $a \in \mathbb{Z}_p$. It trivially holds for $a = 0$. It holds for $a \neq 0$ by Exercise B.41.a (by multiplying both sides of $a^{p-1} = 1$ by a). More generally, $a^d = a^{d-(p-1)}$ holds for every $a \in \mathbb{Z}_p$ and every $d \geq p$.

For any multivariate polynomial over $\mathbb{Z}_p$, we can repeatedly find a monomial with a variable x_i raised to a power $d \geq p$, and replace x_i^d with $x_i^{d-(p-1)}$ in this monomial, without changing the function. This algorithm will halt since the sum over all monomials of the monomial's total degree goes down in each iteration. When it halts, the resulting polynomial expresses the same function as the original polynomial, but now every variable has individual degree $< p$.

Exercise B.41.c: By Corollary B.33, two different canonical-form polynomials of degree $\leq d$ can't agree on more than d points. In particular, with $d = p - 1$, two different canonical-form polynomials of degree $< p$ can't express the same function (since that would mean agreeing on all p points). Viewing canonical-form polynomials of degree $< p$ as pigeons, and viewing functions from $\mathbb{Z}_p$ to $\mathbb{Z}_p$ as holes, this means no hole gets more than one pigeon. The number of pigeons is p^p by the multiplication principle, since there are p choices for each of the p many coefficients (corresponding to powers $0, 1, \ldots, p-1$). The number of holes is also p^p by the multiplication principle, since there are p choices for the output corresponding to each of the p many possible inputs of a function. Since the number of pigeons equals the number of holes, the pigeonhole principle tells us that each hole gets exactly one pigeon. In other words, each function can be expressed in a unique way as a canonical-form polynomial of degree $< p$.

Exercise B.41.d: For each $y \in \mathbb{Z}_p$, let $f_y : \mathbb{Z}_p \to \mathbb{Z}_p$ be the function such that $f_y(y) = 1$ and $f_y(x) = 0$ for all $x \neq y$. By Exercise B.41.c, f_y is expressed by the polynomial $P_y(x)$ of degree $< p$. Now, consider any function $f : \mathbb{Z}_p^n \to \mathbb{Z}_p$, and define the polynomial:

$$P(x_1, \ldots, x_n) = \sum_{y \in \mathbb{Z}_p^n} f(y_1, \ldots, y_n) \cdot P_{y_1}(x_1) \cdots P_{y_n}(x_n)$$

Then P expresses f since for each $z \in \mathbb{Z}_p^n$, the summand in $P(z)$ corresponding to $y = z$ evaluates to $f(z) \cdot 1 \cdots 1 = f(z)$, and each of the other summands evaluates to 0 (since $P_{y_i}(z_i) = 0$ if $y_i \neq z_i$). Furthermore, each variable x_i has individual degree $< p$ in each summand (degree $< p$ in $P_{y_i}(x_i)$ plus degree 0 in $P_{y_j}(x_j)$ for $j \neq i$) and hence has individual degree $< p$ in P.

We have shown that each function is expressed by some canonical-form polynomial with all individual degrees $< p$. The pigeonhole principle tells us the expression must be unique: Viewing canonical-form polynomials with all individual degrees $< p$ as pigeons, and viewing functions from $\mathbb{Z}_p^n$ to $\mathbb{Z}_p$ as holes, we've shown that every hole gets at least one pigeon, but the numbers of pigeons and holes are both p^{p^n}, so in fact every hole gets only one pigeon.

Exercise B.42: $(x^5 + x^3 + 1) - x^2 \cdot (x^3 + x^2 + 1) = x^4 + x^3 + x^2 + 1$

$(x^4 + x^3 + x^2 + 1) - x \cdot (x^3 + x^2 + 1) = x^2 + x + 1$

$\deg(x^2 + x + 1) < \deg(x^3 + x^2 + 1)$

Thus: $x^5 + x^3 + 1 = (x^2 + x) \cdot (x^3 + x^2 + 1) + (x^2 + x + 1)$

Quotient: $x^2 + x$

Remainder: $x^2 + x + 1$

Exercise B.43: The following recursive subroutine maintains the invariant that if P is a univariate polynomial, then factor(P) returns a factorization of P into irreducible polynomials.

```
factor(P):
    if P is irreducible: return P
    let Q_1 and Q_2 be positive-degree polynomials such that P = Q_1 · Q_2
    return factor(Q_1) · factor(Q_2)
```

The invariant trivially holds for the base cases, when P is irreducible. To see that factor(P) maintains the invariant, assume the invariant holds for factor(Q_1) and factor(Q_2). That means for each $i \in \{1, 2\}$, factor(Q_i) returns a product of irreducible polynomials that equals Q_i. Thus, factor(Q_1) $\cdot$ factor(Q_2) is a product of irreducible polynomials that equals $Q_1 \cdot Q_2 = P$, so the invariant holds for factor(P). The recursion terminates because the argument's degree goes down for each recursive call: Since $\deg(Q_1) \geq 1$ and $\deg(Q_2) \geq 1$ and $\deg(Q_1) + \deg(Q_2) = \deg(P)$, we have $\deg(Q_1) < \deg(P)$ and $\deg(Q_2) < \deg(P)$.

Exercise B.44: View $a_1, \ldots, a_{n+1}$ as columns of an $n \times (n+1)$ matrix A over the field $\mathbb{Z}_2$. By Theorem B.40, there exists a nonzero column vector x such that Ax is all 0s. Defining $I = \{i : x_i = 1\}$, we have $I \neq \emptyset$ and $\bigoplus_{i \in I} a_i = \sum_{i=1}^{n+1} x_i a_i = Ax$ is all 0s.

Exercise B.45: If either v or w is the zero vector, then $(v \cdot w)^2 = 0 = (v \cdot v)(w \cdot w)$. For the rest of the proof, assume both v and w are nonzero, so $v \cdot v > 0$ and $w \cdot w > 0$ (Lemma B.38). Define the vectors $x = (\sqrt{w \cdot w})v$ and $y = (\sqrt{v \cdot v})w$. By Lemma B.37.*(i)*:

$$x \cdot x = (w \cdot w)(v \cdot v) \quad \text{and} \quad x \cdot y = \sqrt{(w \cdot w)(v \cdot v)}(v \cdot w) \quad \text{and} \quad y \cdot y = (v \cdot v)(w \cdot w)$$

Applying Lemma B.38 and Lemma B.37.*(ii)*, and noticing that $x \cdot x = y \cdot y$:

$$0 \leq (x - y) \cdot (x - y) = (x \cdot x) - 2(x \cdot y) + (y \cdot y) = 2\big((y \cdot y) - (x \cdot y)\big)$$

Rearranging this yields $x \cdot y \leq y \cdot y$. Plugging in our expressions for $x \cdot y$ and $y \cdot y$, we have $\sqrt{(v \cdot v)(w \cdot w)}(v \cdot w) \leq (v \cdot v)(w \cdot w)$. Dividing both sides by $\sqrt{(v \cdot v)(w \cdot w)}$ (which is positive) produces $v \cdot w \leq \sqrt{(v \cdot v)(w \cdot w)}$. Squaring both sides yields $(v \cdot w)^2 \leq (v \cdot v)(w \cdot w)$, assuming $v \cdot w \geq 0$. If $v \cdot w < 0$ then replacing v with $-v$ in the argument yields

$$-(v \cdot w) = (-v) \cdot w \leq \sqrt{((-v) \cdot (-v))(w \cdot w)} = \sqrt{(v \cdot v)(w \cdot w)}$$

(using Lemma B.37.*(i)* with $c = -1$), and squaring both sides produces $(v \cdot w)^2 \leq (v \cdot v)(w \cdot w)$.

Exercise B.46.a: All the terms with power at least 2 in the power series for e^x are nonnegative, so dropping them can only make the value go down and results in $1 + x$.

Exercise B.46.b: Since $-x$ is positive, so is e^{-x} since all the terms in the power series are positive. Since $e^x e^{-x} = e^{x-x} = e^0 = 1$, we must also have $e^x > 0$ (since a nonpositive times a positive can't equal 1). Thus, $1 + x \leq 0 < e^x$.

Exercise B.46.c: Each term $\frac{x^i}{i!}$ is smaller in absolute value than the previous term $\frac{x^{i-1}}{(i-1)!}$ since it is obtained by multiplying the previous term by $\frac{x}{i}$, which is less than 1 in absolute value since $|x| < 1 \leq i$. Hence, if we group adjacent pairs in the power series like

$$1 + x + \left(\frac{x^2}{2!} + \frac{x^3}{3!}\right) + \left(\frac{x^4}{4!} + \frac{x^5}{5!}\right) + \cdots$$

then each of the pairs (after the $1 + x$) is nonnegative since the first term in the pair has an even power and is thus nonnegative, while the second term in the pair is negative but of smaller absolute value. Thus dropping all those pairs can only make the value go down and results in $1 + x$.

Exercise B.47: First, note that:

- $e^x \geq 1$ for all $x \geq 0$, as in Exercise B.46.a.
- $e^x > 0$ and $e^x = 1/e^{-x}$ for all x, as in Exercise B.46.b.
- e^x is monotonically increasing because if $x > y$ then $e^x = e^{y+(x-y)} = e^y e^{x-y} \geq e^y$ since $e^y > 0$ and $e^{x-y} \geq 1$.

For $x \leq -1$: We have $e^x \leq e^{-1} = 1/e^1 \leq 1$ and $1 + x + x^2 = (x + 1/2)^2 + 3/4 \geq (-1/2)^2 + 3/4 = 1$.

For $-1 < x < 0$: As in Exercise B.46.c, each term $\frac{x^i}{i!}$ is smaller in absolute value than the previous term $\frac{x^{i-1}}{(i-1)!}$. Hence, if we group adjacent pairs in the power series like

$$1 + x + \tfrac{x^2}{2!} + \left(\tfrac{x^3}{3!} + \tfrac{x^4}{4!}\right) + \left(\tfrac{x^5}{5!} + \tfrac{x^6}{6!}\right) + \cdots$$

then each of the pairs (after the $1 + x + \frac{x^2}{2!}$) is negative since the first term in the pair has an odd power and is thus negative, while the second term in the pair is positive but of smaller absolute value. Thus dropping all those pairs can only make the value go up and results in $1 + x + \frac{x^2}{2!} \leq 1 + x + x^2$.

For $0 \leq x \leq 1$: For each $i \geq 3$, we have $1/(i \cdot (i-1) \cdots 4) \leq 1/((i-3)(i-4) \cdots 1) = 1/(i-3)!$ and thus

$$\begin{aligned}
\textstyle\sum_{i=3}^{\infty} \frac{x^i}{i!} &= \tfrac{x^3}{3!} \textstyle\sum_{i=3}^{\infty} \frac{x^{i-3}}{i(i-1)\cdots 4} \\
&\leq \tfrac{x^3}{3!} \textstyle\sum_{i=3}^{\infty} \frac{x^{i-3}}{(i-3)!} && (x \geq 0) \\
&= \tfrac{x^3}{3!} \textstyle\sum_{i=0}^{\infty} \frac{x^i}{i!} \\
&= \tfrac{x^3}{3!} e^x \\
&\leq \tfrac{x^3}{3!} e^1 && (0 \leq x \leq 1) \\
&\leq \tfrac{x^3}{3!} 3 \\
&\leq \tfrac{x^2}{2}
\end{aligned}$$

and so:

$$e^x = \textstyle\sum_{i=0}^{\infty} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2} + \sum_{i=3}^{\infty} \frac{x^i}{i!} \leq 1 + x + \frac{x^2}{2} + \frac{x^2}{2} = 1 + x + x^2$$

Exercise B.48.a: We have $e^{1/i} \geq 1 + \frac{1}{i} = \frac{i+1}{i}$ for every integer $i \geq 1$ by Exercise B.46. Therefore:

$$e^{1/1+1/2+\cdots+1/(n-2)+1/(n-1)} = e^{1/1} \cdot e^{1/2} \cdots e^{1/(n-2)} \cdot e^{1/(n-1)} \geq \frac{2}{1} \cdot \frac{3}{2} \cdots \frac{n-1}{n-2} \cdot \frac{n}{n-1} = n$$

The result follows by taking the natural log of both sides, since e^x is monotonically increasing as noted in the solution to Exercise B.47.

Exercise B.48.b: We have $e^{1/i} \leq 1 + \frac{1}{i} + \frac{1}{i^2} \leq 1 + \frac{1}{i-1} = \frac{i}{i-1}$ for every integer $i \geq 2$ by Exercise B.47 and the fact that:

$$\frac{1}{i-1} - \frac{1}{i} = \frac{i}{i(i-1)} - \frac{i-1}{i(i-1)} = \frac{1}{i(i-1)} \geq \frac{1}{i^2}$$

Therefore:

$$e^{1/2+1/3+\cdots+1/(n-1)+1/n} = e^{1/2} \cdot e^{1/3} \cdots e^{1/(n-1)} \cdot e^{1/n} \leq \frac{2}{1} \cdot \frac{3}{2} \cdots \frac{n-1}{n-2} \cdot \frac{n}{n-1} = n$$

The result follows by taking the natural log of both sides, since e^x is monotonically increasing as noted in the solution to Exercise B.47.