1. **(15 points)** *A string of numbers is called "pleasant" if it consists of some (possibly zero) number of "0"s and "1"s (in any order) followed by some (possibly zero) number of "22"s, "33"s, "44"s, "5"s, and "6"s (in any order). Let a_n represent the number of pleasant strings of length n. Note that $a_0 = 1$ and $a_1 = 4$.*

   (a) **(5 points)** *Explain why a_n is subject to the recurrence $a_n = 2a_{n-1} + 3a_{n-2} + 2^n$ for $n \geq 2$.*

   A string can be pleasant in any of three ways: it could be a shorter pleasant string with "22", "33", or "44" appended to the end, it could be a shorter pleasant string with "5" or "6" appended to the end, or it could be a string consisting entirely of "0"s and "1"s. If we are trying to build a string of length n (for $n \geq 2$), the first type can be created by taking any of the a_{n-2} pleasant strigns of length $n-2$, and appending one of 3 different things to the end, and thus there are $3a_{n-2}$ strings of the first type; the second type can be created by taking any of the a_{n-1} pleasant strigns of length $n-1$ and appending one of 2 different things to the end, and thus there are $2a_{n-1}$ strings of the second type; and lastly, strings of the third type can be constructed from first principles by selecting a "0" or "1" for each of the n positions, which can be done in 2^n different ways. Thus, there are $3a_{n-2} + 2a_{n-1} + 2^n$ pleasant strings of length n.

   (b) **(5 points)** *Give a closed-form formula for a_n.*

   a_n is subject to a linear nonhomogeneous recurrence relation, so to find its general solution we must find a particular solution to the nonhomogeneous recurrence, and then add to it the general solution to the associated homogeneous recurrence.

   Sicne the nonhomogeneous term is 2^n, we offer as a template for the particular solution the assignment $a_n = C2^n$. Then the recurrence will become:

$$\begin{aligned}
a_n &= 2a_{n-1} + 3a_{n-2} + 2^n \\
C2^n &= 2C2^{n-1} + 3C2^{n-2} + 2^n \\
C \cdot 2^2 \cdot 2^{n-2} &= 2C \cdot 2^1 \cdot 2^{n-2} + 3C2^{n-2} + 2^2 \cdot 2^{n-2} \\
4C &= 4C + 3C + 4 \\
C &= \frac{-4}{3}
\end{aligned}$$

   so a particular solution of the nonhomogeneous recurrence is $a_n = \frac{(-4)2^n}{3}$. Note that this does not satisfy the given initial conditions, so we will need to add in terms from the general solution to the associated homogeneous recurrence.

   Let us consider the associated homogeneous recurrence $b_n = 2b_{n-1} + 3b_{n-2}$. Letting $b_n = \lambda^n$, we find that $\lambda^2 - 2\lambda - 3 = 0$, which has solutions $\lambda = 3$ and $\lambda = -1$. Thus, the general solution to this recurrence is $b_n = k \cdot 3^n + \ell(-1)^n$.

   Adding the general terms from the associated homogeneous linear recurrence to our nonhomogeneous recurrence solution, we see that the general solution to the given recurrence is

$$a_n = \frac{(-4)2^n}{3} + k \cdot 3^n + \ell(-1)^n$$

   and we now wish to determine constants k and ℓ to meet the initial conditions $a_0 = 1$ and $a_1 = 4$. Thus:

$$\begin{aligned}
1 = a_0 &= \frac{-4}{3} + \quad k + \ell \\
4 = a_1 &= \frac{-8}{3} + 3k - \ell
\end{aligned}$$

so $5 = -4 + 4k$, so $k = \frac{9}{4}$; we can now easily find ℓ to be $\frac{1}{12}$, so our final closed form is:

$$a_n = \frac{(-16)2^n + 27 \cdot 3^n + (-1)^n}{12}$$

(c) **(5 points)** *Find a closed form for the ordinary generating function* $\sum_{n=0}^{\infty} a_n z^n$.
Note that if we were to "cheat" and derive our result from the previous part we would get

$$\sum_{n=0}^{\infty} \frac{(-16)2^n + 27 \cdot 3^n + (-1)^n}{12} z^n = \frac{-4}{3(1-2z)} + \frac{9}{4(1-3z)} + \frac{1}{12(1+z)}$$

We can of course derive an equivalent result directly from the recurrence relation as well. For brevity we will let $f(z) = \sum_{n=0}^{\infty} a_n z^n$, and combining the recurrence for each $n \geq 2$ yields the equivalence of power series:

$$\begin{aligned}
\sum_{n=2}^{\infty} a_n z^n &= \sum_{n=2}^{\infty} (2a_{n-1} + 3a_{n-2} + 2^n) z^n \\
\sum_{n=2}^{\infty} a_n z^n &= 2z \left(\sum_{n=2}^{\infty} a_{n-1} z^{n-1} \right) + 3z^2 \left(\sum_{n=2}^{\infty} a_{n-2} \right) + \sum_{n=2}^{\infty} (2z)^n \\
f(z) - 4z - 1 &= 2z (f(z) - 1) + 3z^2 f(z) + \frac{1}{1-2z} - 2z - 1 \\
f(z) - 2zf(z) - 3z^2 f(z) &= \frac{1}{1-2z} \\
f(z) &= \frac{1}{(1-2z)(1-2z-3z^2)} = \frac{1}{(1-2z)(1-3z)(1+z)}
\end{aligned}$$

Now, if we had not already worked out what a_n was in part (b), we could derive it from this result, by partial-fraction decomposition:

$$\frac{1}{(1-2z)(1-3z)(1+z)} = \frac{A}{1-2z} + \frac{B}{1-3z} + \frac{C}{1+z}$$

which would require solution of the polynomial equivalence

$$0z^2 + 0z + 1 \equiv (-3z^2 - 2z + 1)A + (-2z^2 - z + 1)B + (6z^2 - 5z + 1)C$$

or in other words, the system of equations

$$\begin{aligned}
-3A - 2B + 6C &= 0 \\
-2A - B - 5C &= 0 \\
A + B + C &= 1
\end{aligned}$$

which indeed has solution $A = \frac{-4}{3}$, $B = \frac{9}{4}$, and $C = \frac{1}{12}$, so

$$f(z) = \frac{-4}{3(1-2z)} + \frac{9}{4(1-3z)} + \frac{1}{12(1+z)} = \sum_{n=0}^{\infty} \frac{(-16)2^n + 27 \cdot 3^n + (-1)^n}{12} z^n$$

which agrees with results we saw previously derived by other methods.

2. **(10 points + 5 point bonus)** *I have chosen an integer between 1 and n, and when you guess a number, I will tell you whether my number is higher, lower, or equal.*

   (a) **(10 points)** *Specifically describe, in step-by-step detail, a procedure to figure out what my number is with $\lceil \log_2 n \rceil$ or fewer guesses.*

   You start by setting the "lower limit" of the possible numbers to be 1 and "upper limit" to be n.

   Now you repeat the following procedure until victorious: you guess a number which is the average (rounded down to an integer, if necessary) of the upper and lower limits, and respond as such: if my number is equal, you declare victory; if my number is smaller, you set the "upper limit" to be one less than the number you guessed; if my number is larger, you set the "lower limit" to be one more than the number you guessed. If the upper and lower limits are equal, you can declare victory since that must be my number; otherwise, you return to the beginning of this paragraph and continue.

   The argument that this will use $\lceil \log_2 n \rceil$ or fewer guesses is pretty simple. Let us denote the number of guesses needed to determine the secret where the lower limit is a and the upper limit is b by $f(b-a)$. Clearly $f(0) = 0$, since when $a = b$ we declare victory immediately. The method described above shows that for $b - a \geq 1$, we guess $\lfloor \frac{a+b}{2} \rfloor = \lfloor a + \frac{b-a}{2} \rfloor$, after which one of three things happens: either we are done, or we need to further narrow down a secret number from a to $\lfloor a + \frac{b-a}{2} \rfloor - 1$, or we need to further narrow down a secret number from $\lfloor a + \frac{b-a}{2} \rfloor + 1$ to b. Thus, since $f(b-a)$ represents the number of guesses needed in the worst case:

$$f(b-a) \leq 1 + \max(0, f\left(\lfloor a + \frac{b-a}{2} \rfloor - 1 - a\right), f\left(b - (\lfloor a + \frac{b-a}{2} \rfloor + 1)\right))$$

   and since f is clearly an increasing nonnegative function, we can be certain the last of these is the largest:

$$f(b-a) \leq 1 + f\left(\lceil \frac{b-a}{2} \rceil - 1\right)$$

   We note that $\lceil \frac{b-a}{2} \rceil - 1 < \frac{b-a}{2}$, so $f(b-a)$ is one more than the number of guesses necessary to find a number in an interval whose length is less than half of $b - a$. Thus, if $b - a < 2^k$, then invoking this rule k times we see that $f(b-a) \leq k + f(\epsilon)$, where $\epsilon < \frac{b-a}{2^k} < 1$, so $\epsilon = 0$ and thus $f(b-a) \leq k$, and by definition $k > \log_2(b-a)$, so in the specific case of $b = n$ and $a = 1$, we get that $f(n-1) < \log_2(n-1)$.

   (b) **(5 point bonus)** *Explain why, in general, it would be impossible to guess my number in fewer than $\lceil \log_2(n+1) \rceil - 1$ guesses.*

   Let us suppose, contrariwise, that there is some value n and algorithm thereon which can guess the number in fewer than $\lceil \log_2(n+1) \rceil - 1$ guesses. Let n_0 be the smallest integer such that there is an algorithm which can always guess a secret integer from 1 to n_0 in $\lceil \log_2(n_0+1) \rceil - 2$ or fewer steps. Note that $\lceil \log_2(2+1) \rceil - 2 < 0$, so it is a certainty that $n_0 > 2$.

   Suppose the algorithm searching through the interval $[1, n_0]$ starts with some guess x. There are three possibilities: either x was correct, or x was too high, or x was too low. The first case is trivial; in the latter two cases, our algorithm must be able to find the secret in the remaining $\lceil \log_2(n_0+1) \rceil - 3$ steps. In the first case our algorithm is searching through $[1, x-1]$; in the second case it is searching through $[x+1, n_0]$, which could be translated to be a search through the identical interval $[1, n_0 - x]$. Let $y = \max(x-1, n_0 - x)$. Since

$2y \geq (x-1)+(n_0-x) = n_0-1$, we know that $y \geq \frac{n_0-1}{2}$; we also know, since $1 \leq x \leq n_0$, that $y < n_0$ We asserted earlier that we must be able to find a secret in the range $[1, y]$ in $\lceil \log_2(n_0+1) \rceil - 3$ steps; however, note that

$$\lceil \log_2(y+1) \rceil - 2 \geq \lceil \log_2(\frac{n_0-1}{2}+1) \rceil - 2 = \lceil \log_2(n_0+1) \rceil - 3$$

so y violates our definition of n_0, as it is smaller than n_0 but there is an algorithm which can always guess a secret integer from 1 to y in $\lceil \log_2(y+1) \rceil - 2$ or fewer steps; thus our premise that such an n_0 exists is incorrect.

3. **(15 points)** *The binary form of a number is its value written out in base 2. For instance, 37 has binary form 100101, because it is* $1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$*. In the following question, outputting a 0 or 1 is a single operation.*

   (a) **(5 points)** *Write, in detail, a procedure which, given a number n, will output its binary form. If you find it easier, you may choose to write the number backwards: i.e. for 37, you could also write a procedure which outputs 101001.*

   It is worth noting that the final bit is the remainder when the number is divided by 2; the penultimate bit is the remainder when the result of that division is divided by 2, the previous bit is the result of *that* division by 2, and so on. We can thus write a simple algorithm encapsulating this idea; note that this will actually output the bits backwards.

   i. If n is odd, output "1"; otherwise, output "0".
   ii. Divide n by 2, rounding down; assign the resulting value to n.
   iii. If $n = 0$, then terminate the procedure.
   iv. Return to step i.

   If we wanted to write the number in unreversed binary order, we could do that too, with the use of a recursive procedure.

   i. Let y be equal to $\lfloor \frac{n}{2} \rfloor$.
   ii. If $y > 0$, then using a recursive call, write y in binary.
   iii. If n is odd, output "1"; otherwise, output "0".

   (b) **(10 points)** *How many operations will your procedure use to write out a number n? Explain your reasoning.*

   Since either version of this procedure uses a constant number of steps to reducing the problem to writing $\lfloor \frac{n}{2} \rfloor$ in binary, we see that after k iterations, we will be reduced to the problem of writing $\lfloor \frac{n}{2^k} \rfloor$ in binary. If $k = \lceil \log_2 n \rceil$, then $0 \leq \lfloor \frac{n}{2^k} \rfloor \leq 1$, which is written in binary in a single step. Thus, we expect either the recursive or looping version of this procedure to terminate after $\lceil \log_2 n \rceil$ iterations, each iteration consisting of a constant number of steps, so this uses $O(\log n)$ operations.

4. **(10 points)** *Explain why every simple graph must contain two vertices of the same degree.*

   Suppose G is a graph with n vertices. Degrees are definitionally non-negative, so the degrees of the vertices of G are no smaller than 0; since each vertex can have at most $n-1$ neighbors, the largest degree is $n-1$. It is clearly impossible for there to be both a vertex of degree 0 *and* a vertex of degree $n-1$ in the graph, however: the former vertex would be adjacent to no other vertex, and the latter vertex would be adjacent to every other vertex, so the edge between them would need to be simultaneously present and absent. Thus, a graph on n vertices can attain at most $n-1$ distinct degrees on its vertices ($0, 1, 2, \ldots, n-2$ and at most one of 0 or $n-1$), and by the pigeonhole principle, since $n > n-1$, this means the same degree must be used twice.

5. **(10 points)** *Let the vertices of a graph G have degree between δ and Δ. Show that $\delta \leq \frac{2\|G\|}{|G|} \leq \Delta$.*
For each vertex v of G, we are told that $\delta \leq d(v) \leq \Delta$. Adding up these inequalities over every vertex v:

$$\sum_{v \in V(G)} \delta \leq \sum_{v \in V(G)} d(v) \leq \sum_{v \in V(G)} \Delta$$
$$|G|\delta \leq 2\|G\| \leq |G|\Delta$$
$$\delta \leq \frac{2\|G\|}{|G|} \leq \Delta$$

Since $\frac{2\|G\|}{|G|} \frac{\sum_{v \in V(G)} d(v)}{|V(G)|}$, which is the arithmetic average over the vertices of their degrees, it is often called the *average degree*; conceptually it is not a surprise that the average degree is between the minimum and maximum degrees.

6. **(5 point bonus)** *The complement G^c of a simple graph G is a graph on the same vertex-set as G, but such that every pair of adjacent vertices is non-adjacent in G^c, and vice versa. A graph is called* self-complement *if it is isomorphic to its own complement; for instance, the cycle C_5 is self-complement. Explain why, if G is self-complement, it must be the case that $|G|$ is either a multiple of 4 or one more than an multiple of 4.*
Let $|G| = n$, and suppse G is self-complement. Since every possible pair of distinct vertices are adjacent in exactly one of G or G^c, it follows that

$$\|G\| + \|G^c\| = \binom{n}{2}$$

And since G and G^c are isomorphic, $\|G^c\| = \|G\|$, so

$$2\|G\| = \frac{(n)(n-1)}{2}$$

and thus it must be the case that $n(n-1) = 4\|G\|$; since $\|G\|$ is an integer, it is thus the case that the product $n(n-1)$ is a multiple of 4; since exactly one of n or $n-1$ is odd, the other must be a multiple of 4.
There are in fact self-complementary graphs on every number of vertices of the form $4k$ or $4k+1$; the proof of this statement is not simple, however.

Die ganzen Zahlen hat der liebe Gott gemacht: alles andere ist Menschenwerk. [The good Lord made the natural numbers; all else is the work of man.] —attributed to Leopold Kronecker