> The problems in this assignment are *tagged*:
>
> Core are essential problems that you *must solve* to pass the assignment.
>
> Challenge are *non-mandatory* problems that could be *skipped* for a passing grade.
>
> Bonus are *optional* problems for *memorable experience*. They won't be graded.

Problem 1: Solving Recurrences

Core

For each given recurrence relation, find the first five terms, derive the closed-form solution, and check it by substituting it back to the recurrence relation.

(a) $a_n = a_{n-1} + n$ with $a_0 = 2$

(b) $a_n = 2a_{n-1} + 2$ with $a_0 = 1$

(c) $a_n = 3a_{n-1} + 2^n$ with $a_0 = 5$

(d) $a_n = 4a_{n-1} + 5a_{n-2}$ with $a_0 = 1, a_1 = 17$

(e) $a_n = 4a_{n-1} - 4a_{n-2}$ with $a_0 = 3, a_1 = 11$

(f) $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$ with $a_{0,1,2} = 3, 2, 6$

Problem 2: Master Theorem and Akra–Bazzi

Core

Solve the following recurrences by applying the Master theorem. For the cases where the Master theorem does not apply, use the Akra–Bazzi method. In cases where neither of these two theorems apply, explain why and solve the recurrence relation by closely examining the recursion tree. Solutions must be in the form $T(n) \in \Theta(\ldots)$.

(a) $T(n) = 2T(n/2) + n$

(b) $T(n) = T(3n/4) + T(n/4) + n$

(c) $T(n) = 3T(n/2) + n$

(d) $T(n) = 2T(n/2) + n/\log n$

(e) $T(n) = 6T(n/3) + n^2 \log n$

(f) $T(n) = T(3n/4) + n \log n$

(g) $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n$

(h) $T(n) = T(n/2) + T(n/4) + 1$

(i) $T(n) = T(n/2) + T(n/3) + T(n/6) + n$

(j) $T(n) = 2T(n/3) + 2T(2n/3) + n$

(k) $T(n) = \sqrt{2n}T\left(\sqrt{2n}\right) + \sqrt{n}$

(l) $T(n) = \sqrt{2n}T\left(\sqrt{2n}\right) + n$

Problem 3: Estimating Square Roots via Recurrences

Challenge

Consider a recurrence relation $a_n = 2a_{n-1} + 2a_{n-2}$ with $a_0 = a_1 = 2$. Solve it (*i.e.* find a closed formula) and show how it can be used to estimate the value of $\sqrt{3}$ (hint: observe $\lim_{n\to\infty} a_n/a_{n-1}$).

After that, devise an algorithm for constructing a recurrence relation with integer coefficients and initial conditions that can be used to estimate the square root $\sqrt{k}$ of a given integer k.

Problem 4: Generating Function — Closed Formula

Core

Find a closed formula for the n-th term of the sequence with generating function

$$\frac{3x}{1-4x} + \frac{1}{1-x}$$

Problem 5: Partial Fractions Decomposition

Core

Given the generating function $G(x) = \frac{5x^2+2x+1}{(1-x)^3}$, decompose it into partial fractions and find the sequence that it represents.

Problem 6: Pell–Lucas Numbers

Core

Pell–Lucas numbers are defined by $Q_0 = Q_1 = 2$ and $Q_n = 2Q_{n-1} + Q_{n-2}$ for $n \geq 2$.

Derive the corresponding generating function and find a closed formula for the n-th Pell–Lucas number.

Problem 7: Generating Functions for Recurrences

Core

For each given recurrence relation, derive the corresponding generating function and find a closed formula for the n-th term of the sequence.

(a) $a_n = 2a_{n-1} - a_{n-2}$ with $a_0 = 3$, $a_1 = 5$
(b) $a_n = a_{n-1} + a_{n-2} - a_{n-3}$ with $a_0 = 1$, $a_1 = 1$, $a_2 = 5$
(c) $a_n = a_{n-1} + n$ with $a_0 = 0$
(d) $a_n = a_{n-1} + 2a_{n-2} + 2^n$ with $a_0 = 2$, $a_1 = 1$

Problem 8: Diophantine Equation via Generating Functions

Core

Find the number of non-negative integer solutions to the Diophantine equation

$$3x + 5y = 100$$

using generating functions.

Problem 9: Lucky Tickets

Challenge

Consider a $2n$-digit ticket number to be *lucky* if the sum of its first n digits is equal to the sum of its last n digits. Each digit (including the first one!) in a number can take value from 0 to 9. For example, a 6-digit ticket $345,264$ is lucky since $3 + 4 + 5 = 2 + 6 + 4$.

(a) Find the number of lucky 6-digit and 8-digit tickets.
(b) Find the generating function for the number of $2n$-digit lucky tickets.
(c) Find a closed formula for the number of $2n$-digit lucky tickets.

Optional Bonus Problems

> The following problems are *"optional"*. They require programming and/or deeper theory. They will *not* count toward your grade but *may* earn bonus points and genuine understanding.

Problem A: The Fibonacci Zoo

Bonus

In 1202, Leonardo of Pisa — known as *Fibonacci* — posed a simple question about rabbit breeding. The resulting sequence $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$ appears in sunflower spirals, pinecone bracts, and the worst-case height of AVL trees. But beneath the familiar $F_n = (\varphi^n - \psi^n)/\sqrt{5}$ lies a zoo of surprising phenomena.

> ***Fibonacci recurrence.*** $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ *for* $n \geq 2$. *Binet:* $F_n = \frac{\varphi^n - \psi^n}{\sqrt{5}}$ *where* $\varphi = \frac{1+\sqrt{5}}{2}$, $\psi = \frac{1-\sqrt{5}}{2}$.

1. **Pisano periods.** Reduce every Fibonacci number modulo m and the remainders *always become periodic*. This period $\pi(m)$ is the *Pisano period* — nobody has a closed formula for it.
   - Compute $\pi(m)$ for $m = 2, 3, 4, 5, 6, 7, 8, 9, 10, 100$ by brute force.
   - Prove: if p is prime and $p \equiv \pm 1 \pmod 5$, then $\pi(p)$ divides $p - 1$. (Apply Fermat's little theorem to Binet's formula.)
   - The last digit of F_n repeats every 60 terms. Explain why $\pi(10) = \operatorname{lcm}(\pi(2), \pi(5))$ and compute $\pi(2)$, $\pi(5)$.
2. **Zeckendorf: every number is a sum of non-consecutive Fibonaccis.** Zeckendorf (1939): *every* positive integer has a *unique* representation as a sum of non-consecutive Fibonacci numbers (from $F_2 = 1$). For example, $100 = 89 + 8 + 3 = F_{11} + F_6 + F_4$.
   - Prove that the greedy algorithm (always take the largest $F_k \leq N$, subtract, repeat) never selects two consecutive Fibonacci numbers. (If it picks F_k and F_{k-1}, then $F_k + F_{k-1} = F_{k+1}$ — contradiction.)
   - Prove uniqueness: two different non-consecutive subsets of $\{F_2, F_3, \ldots\}$ cannot sum to the same N.
   - Implement the greedy algorithm. Write the Zeckendorf representation of 42, 100, and 2025.
3. **Fibonacci coding: turning Zeckendorf into a prefix code.** Write the Zeckendorf representation in reverse binary (bit i = 1 iff F_{i+2} is used), then append an extra `1`. The `11` suffix is a unique terminator — no codeword is a prefix of another, so you can concatenate messages without delimiters.
   - Encode `"HELLO"` ($A \to 1, \ldots, Z \to 26$) using Fibonacci coding. Compare total bits with $5 \times 5 = 25$ bits of fixed-width encoding.
   - Prove prefix-freeness: every codeword ends in `11` and no Zeckendorf representation has consecutive 1-bits.

Problem B: Euler's Pentagonal Magic

Bonus

In 1741, Euler wrote to Berlin claiming an impossible identity: the infinite product $(1 - x)(1 - x^2)(1 - x^3)\cdots$ should have coefficients everywhere, but when expanded, *almost all vanish* — the only nonzero ones sit at the positions $1, 2, 5, 7, 12, 15, 22, 26, \ldots$, known as *pentagonal numbers*

$k(3k-1)/2$. From this single identity, Euler derived a recurrence for $p(n)$, the number of partitions of n. A century and a half later, Ramanujan discovered that $p(5k+4)$ is always divisible by 5, and $p(7k+5)$ by 7. This problem follows the thread from Euler's product to Ramanujan's congruences.

Euler's pentagonal number theorem.

$$\prod_{k=1}^{\infty}(1-x^k) = \sum_{k=-\infty}^{\infty}(-1)^k x^{k(3k-1)/2} = 1 - x - x^2 + x^5 + x^7 - x^{12} - x^{15} + \dots$$

Partition generating function.

$$\sum_{n=0}^{\infty} p(n)x^n = \prod_{k=1}^{\infty}\frac{1}{1-x^k}$$

where $p(n)$ counts the ways to write n as a sum of positive integers in non-increasing order.

1. **Vanishing coefficients and the partition GF.** Expand $(1-x)(1-x^2)\cdots(1-x^{30})$ and inspect the coefficients: they are only $0, +1, -1$, and the nonzero indices form the sequence $1, 2, 5, 7, 12, 15, 22, 26, \dots$
   - Verify that these are exactly $k(3k-1)/2$ for $k = \pm1, \pm2, \pm3, \dots$ and that the sign at index $k(3k-1)/2$ is $(-1)^k$. Does the coefficient of x^n vanish for *every* n that is *not* a pentagonal number?
   - The partition GF is $\sum_{n=0}^{\infty} p(n)x^n = \prod_{k=1}^{\infty}\frac{1}{1-x^k}$. Why? Each factor $\frac{1}{1-x^k} = 1 + x^k + x^{2k} + \dots$ counts how many times part k appears. Multiply 20 factors and read off $p(1)$ through $p(20)$. Verify: $p(5) = 7$.

2. **Euler's recurrence for $p(n)$.** Multiply the pentagonal theorem by the partition GF:

$$1 = \left(\sum_{k}(-1)^k x^{k(3k-1)/2}\right)\cdot\left(\sum_{n} p(n)x^n\right)$$

   For $n \geq 1$, the coefficient of x^n on the right must vanish. Setting it to zero gives a recurrence with pentagonal-number offsets:

$$p(n) = p(n-1) + p(n-2) - p(n-5) - p(n-7) + p(n-12) + p(n-15) - \dots$$

   (signs alternate in pairs, negative indices treated as zero).
   - Implement this recurrence and compute $p(n)$ for $n = 0, \dots, 100$. Verify $p(10) = 42$, $p(50) = 204226$, $p(100) = 190569292$.

3. **Ramanujan's congruences.** Ramanujan proved: $p(5k+4) \equiv 0 \pmod 5$, $p(7k+5) \equiv 0 \pmod 7$, $p(11k+6) \equiv 0 \pmod{11}$. Verify all three for $k = 0, 1, \dots, 50$ using your recurrence.

4. **The Hardy–Ramanujan asymptotic.** Hardy and Ramanujan (1918) proved: $p(n) \sim \frac{1}{4n\sqrt{3}}\exp\left(\pi\sqrt{2n/3}\right)$. Compute $p(n)$ exactly via the recurrence and compare with the asymptotic formula for $n = 5, 20, 50, 100, 500, 1000$. Plot the relative error. How many digits does the asymptotic get right for $p(1000)$?

Problem C: Linear Recurrence Solver

Bonus

You've solved recurrences by hand — characteristic polynomials, partial fractions, careful algebra. Now build a *tool* that does it automatically: from recurrence to closed form, and from closed form to a_n for $n = 10^{18}$ in milliseconds.

Companion matrix. *$a_n = c_1 a_{n-1} + \ldots + c_k a_{n-k}$ becomes*

$$(\boldsymbol{v}_n) = \begin{pmatrix} c_1 & c_2 & \ldots & c_k \\ 1 & 0 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 1 & 0 \end{pmatrix} (\boldsymbol{v}_{n-1})$$

Repeated squaring computes a_n in $O(k^3 \log n)$ time.

1. **Characteristic polynomial pipeline.** Given a recurrence and initial conditions, produce the closed form: (1) characteristic polynomial from coefficients, (2) all roots (complex, repeated, or both), (3) solve a linear system for coefficients from initial conditions, (4) assemble $a_n = \sum_i p_i(n) r_i^n$ where p_i has degree one less than the multiplicity of r_i. Test on every recurrence from Problem 1 — verify the first 10 terms.

2. **Matrix exponentiation: when you need the number, not the formula.** Implement repeated squaring and compute:
   - F_{100} (Fibonacci). Verify.
   - $F_{10^{18}} \bmod (10^9 + 7)$. If your code runs more than a second, something is wrong.
   - Problem 1(d): $a_n = 4a_{n-1} + 5a_{n-2}$, $a_0 = 1, a_1 = 17$. Compute $a_{10^{15}} \bmod (10^9 + 7)$. Benchmark against $O(n)$ iteration — where is the crossover?

3. **Applications.** Find a recurrence for the number of domino tilings of a $3 \times n$ board (compute small cases, spot the pattern), then use your solver for the closed form.

   Also compute $T_{10^{12}} \bmod (10^9 + 7)$ for the Tribonacci sequence: $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ with $T_0 = 0, T_1 = 0, T_2 = 1$.