International Islamic University Chittagong

IIUC_MARK_US

Arman, Istiaque, Mizan

Mathematics

Equations

The extremum of a quadratic is given by $x = -b/2a$.
Cramer's Rule: Given an equation $Ax = b$, the solution to a variable x_i is given by

$$x_i = \frac{\det A'_i}{\det A}$$ [where A'_i is A with the i'th column replaced by b.]

Example (3x3):

$$\begin{aligned} 2x + 3y - 5z &= 1 \\ x + y - z &= 2 \\ 2y + z &= 8 \end{aligned}$$

$D = \begin{vmatrix} 2 & 3 & -5 \\ 1 & 1 & -1 \\ 0 & 2 & 1 \end{vmatrix} = -7$ $D_x = \begin{vmatrix} 1 & 3 & -5 \\ 2 & 1 & -1 \\ 8 & 2 & 1 \end{vmatrix} = -7$ $D_y = \begin{vmatrix} 2 & 1 & -5 \\ 1 & 2 & -1 \\ 0 & 8 & 1 \end{vmatrix} = -21$ $D_z = \begin{vmatrix} 2 & 3 & 1 \\ 1 & 1 & 2 \\ 0 & 2 & 8 \end{vmatrix} = 14$ $x = \frac{D_x}{D} = 1$, $y = \frac{D_y}{D} = 3$, $z = \frac{D_z}{D} = -2$

Vieta's Formulas: Let $P(x) = a_n x^n + ... + a_0$, be a polynomial with complex coefficients and degree n, having complex roots $r_n, ..., r_1$. Then for any integer $0 \le k \le n$,

$$\sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} r_{i_1} r_{i_2} ... r_{i_k} = (-1)^k \frac{a_{n-k}}{a_n}$$

Rational Root Theorem: If $\frac{p}{q}$ is a reduced rational root of a polynomial with **integer coeffs**, then $p \mid a_0$ and $q \mid a_n$.

Number Theory

Sum of Divisors (S.O.D): If $N = a^p \cdot b^q \cdot c^r \ldots$

$$\text{S.O.D} = \frac{a^{p+1}-1}{a-1} \cdot \frac{b^{q+1}-1}{b-1} \cdot \frac{c^{r+1}-1}{c-1} \ldots$$

Number of Divisors (N.O.D): If $N = a^p \cdot b^q \cdot c^r \ldots$

$$\text{N.O.D} = (p+1)(q+1)(r+1)\ldots$$

Product of Divisors (P.O.D): If N has $D = \text{N.O.D}(N)$ divisors:

$$\text{P.O.D}(N) = N^{D/2} = (\sqrt{N})^D$$

Euclidean Algorithm Property:

$$\gcd(a,b) = \gcd(a, a-b) \quad [a > b]$$

Fibonacci GCD:

$$\gcd(F(a), F(b)) = F(\gcd(a,b))$$

Euler's Totient Theorem:

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

where $\phi(n)$ is Euler's Totient Function.
Modular Exponentiation:

$$a^b \pmod{m} \equiv a^{b \pmod{\phi(m)}} \pmod{m}$$

(if a and m are coprime)
Primitive roots modulo n exists iff $n = 1, 2, 4$ or, $n = p^k, 2p^k$ where p is an odd prime. Furthermore, the number of roots are $\phi(\phi(n))$.
To Find Generator g of M, factor $M-1$ and get the distinct primes p_i. If $g^{(M-1)/p_i} \neq 1 (MODM)$ for each p_i then g is a valid root. Try all g until a hit is found (usually found very quick).

Mobius Function

$$\mu(n) = \begin{cases} 0 & n \text{ is not square free} \\ 1 & n \text{ has even number of prime factors} \\ -1 & n \text{ has odd number of prime factors} \end{cases}$$

Mobius Inversion:

$$g(n) = \sum_{d|n} f(d) \Leftrightarrow f(n) = \sum_{d|n} \mu(d) g(n/d)$$

Other useful formulas/forms:
$\sum_{d|n} \mu(d) = [n = 1]$, $\phi(n) = \sum_{d|n} \mu(d) \frac{n}{d}$
$g(n) = \sum_{n|d} f(d) \Leftrightarrow f(n) = \sum_{n|d} \mu(\frac{d}{n}) g(d)$
$g(n) = \sum_{1 \le m \le n} f(\lfloor \frac{n}{m} \rfloor) \Leftrightarrow f(n) = \sum_{1 \le m \le n} \mu(m) g(\lfloor \frac{n}{m} \rfloor)$
If f multiplicative, $\sum_{d|n} \mu(d) f(d) = \prod_{\text{prime } p|n} (1 - f(p))$ and $\sum_{d|n} \mu^2(d) f(d) = \prod_{\text{prime } p|n} (1 + f(p))$.
If $s_f(n) = \sum_{i=1}^{n} f(i)$ is a prefix sum of mulitplicative f then $s_{f*g}(n) = \sum_{1 \le xy \le n} f(x) g(y)$. Then $s_f(n) = \{s_{f*g}(n) - \sum_{d=2}^{n} s_f(\lfloor n/d \rfloor) g(d)\} / g(1)$ where $f * g(n) = \sum_{d|n} f(d) g(n/d)$ (Dirichlet). Precompute (linear sieve) $O(n^{2/3})$ first values of s_f for complexity $O(n^{2/3})$.
Useful sums and convolutions: $\epsilon = \mu * \mathbf{1}$, $\text{id} = \phi * \mathbf{1}$, $\text{id} = g * \text{id}_2$, where $\epsilon(n) = [n = 1]$, $\mathbf{1}(n) = 1$, $\text{id}(n) = n$, $\text{id}_k(n) = n^k$, $g(n) = \sum_{d|n} \mu(d) nd$.
coprime pairs in $[1, n]$ is $\sum_{d=1}^{n} \mu(d) \lfloor n/d \rfloor^2$. Sum of GCD pairs in $[1, n]$ is $\sum_{d=1}^{n} \phi(d) \lfloor n/d \rfloor^2$. Sum of LCM pairs in $[1, n]$ is $\sum_{d=1}^{n} (\frac{\lfloor n/d \rfloor (1 + \lfloor n/d \rfloor)}{2})^2 g(d)$, where g is defined above with $g(p^k) = p^k - p^{k+1}$.

Partition Function: $p(n)$

Pattern: Form a sum n where the **order does not matter**.

- "How many ways to write n as a sum of positive integers?"
- "How many ways to put n *identical* balls into *identical* boxes?"

Definition: Number of ways of writing n as a sum of positive integers, disregarding order. **Sequence** $p(n)$ **for** $n = 0, 1, 2, \ldots$**:**

$$1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, 56, 77, \ldots$$

Recurrence (Pentagonal Number Theorem):

$$\begin{aligned} p(n) &= \sum_{k \in \mathbb{Z} \setminus \{0\}} (-1)^{k-1} p(n - k(3k-1)/2) \\ &= p(n-1) + p(n-2) - p(n-5) - p(n-7) + \ldots \end{aligned}$$

Ceils and Floors

For $x, y \in \mathbb{R}$, $m, n \in \mathbb{Z}$:

- $\lfloor x \rfloor \le x < \lfloor x \rfloor + 1$; $\lceil x \rceil - 1 < x \le \lceil x \rceil$
- $-\lfloor x \rfloor = \lceil -x \rceil$; $-\lceil x \rceil = \lfloor -x \rfloor$
- $\lfloor x + n \rfloor = \lfloor x \rfloor + n$, $\lceil x + n \rceil = \lceil x \rceil + n$
- $\lfloor x \rfloor = m \Leftrightarrow x - 1 < m \le x < m + 1$
- $\lceil x \rceil = n \Leftrightarrow n - 1 < x \le n < x + 1$
- If $n > 0$, $\lfloor \frac{\lfloor x \rfloor + m}{n} \rfloor = \lfloor \frac{x+m}{n} \rfloor$
- If $n > 0$, $\lceil \frac{\lceil x \rceil + m}{n} \rceil = \lceil \frac{x+m}{n} \rceil$
- If $n > 0$, $\lfloor \frac{\lfloor \frac{x}{m} \rfloor}{n} \rfloor = \lfloor \frac{x}{mn} \rfloor$
- If $n > 0$, $\lceil \frac{\lceil \frac{x}{m} \rceil}{n} \rceil = \lceil \frac{x}{mn} \rceil$
- For $m, n > 0$, $\sum_{k=1}^{n-1} \lfloor \frac{km}{n} \rfloor = \frac{(m-1)(n-1) + \gcd(m,n) - 1}{2}$
- $\lfloor n/j \rfloor = x$ for $j \in [\lfloor n/(x+1) \rfloor + 1, \lfloor n/x \rfloor]$
- Modulo definition: $a \pmod{m} = a - m \lfloor a/m \rfloor$

Recurrences

If $a_n = c_1 a_{n-1} + \cdots + c_k a_{n-k}$, and $r_1, \ldots, r_k$ are distinct roots of $x^k - c_1 x^{k-1} - \cdots - c_k$, there are $d_1, \ldots, d_k$ s.t.

$$a_n = d_1 r_1^n + \cdots + d_k r_k^n.$$

Non-distinct roots r become polynomial factors, e.g. $a_n = (d_1 n + d_2) r^n$.

Trigonometry

$$\sin(v + w) = \sin v \cos w + \cos v \sin w$$
$$\cos(v + w) = \cos v \cos w - \sin v \sin w$$
$$\tan(v + w) = \frac{\tan v + \tan w}{1 - \tan v \tan w}$$
$$\sin v + \sin w = 2 \sin \frac{v + w}{2} \cos \frac{v - w}{2}$$
$$\cos v + \cos w = 2 \cos \frac{v + w}{2} \cos \frac{v - w}{2}$$

$$(V + W) \tan(\frac{v - w}{2}) = (V - W) \tan(\frac{v + w}{2})$$

V, W are sides opposite to angles v, w. $a \cos x + b \sin x = r \cos(x - \phi)$
$a \sin x + b \cos x = r \sin(x + \phi)$
where $r = \sqrt{a^2 + b^2}, \phi = \text{atan2}(b, a)$.

Geometry

Rectangles and Squares

- Area of a rectangle: $A = l \cdot w$
- Perimeter of a rectangle: $P = 2l + 2w$
- Diagonal of a rectangle: $d = \sqrt{l^2 + w^2}$
- Area of a square: $A = \text{side}^2$
- Perimeter of a square: $P = 4 \cdot \text{side}$
- Diagonal of a square: $d = \sqrt{2} \cdot \text{side}$

Triangles

Side lengths: a, b, c; Semiperimeter: $p = \frac{a + b + c}{2}$

- Area: $A = \frac{1}{2} \cdot b \cdot h$
- Perimeter: $P = a + b + c$
- Heron's Area: $A = \sqrt{p(p-a)(p-b)(p-c)}$
- Circumradius: $R = \frac{abc}{4A}$
- Inradius: $r = \frac{A}{p}$
- Length of median: $m_a = \frac{1}{2} \sqrt{2b^2 + 2c^2 - a^2}$
- Length of bisector: $s_a = \sqrt{bc \left[1 - (a/(b+c))^2\right]}$
- Law of Sines: $\frac{\sin \alpha}{a} = \frac{\sin \beta}{b} = \frac{\sin \gamma}{c} = \frac{1}{2R}$
- Law of Cosines: $a^2 = b^2 + c^2 - 2bc \cos \alpha$
- Law of Tangents: $\frac{a + b}{a - b} = \frac{\tan((\alpha + \beta)/2)}{\tan((\alpha - \beta)/2)}$

Circles

- Area: $A = \pi \cdot r^2$
- Circumference: $C = 2\pi \cdot r$
- Sector Area: $A_{\text{sector}} = \frac{\theta}{360°} \cdot \pi \cdot r^2$ (in degrees)
- Arc Length: $l = \frac{\theta}{360°} \cdot 2\pi \cdot r$ (in degrees)

Polygons (n-sided)

- Sum of interior angles: $(n - 2) \times 180°$
- A single angle (regular): $\frac{(n-2) \times 180°}{n}$
- Amount of diagonals: $\frac{n(n-3)}{2}$
- Sum of exterior angles: $360°$
- Area (regular): $\frac{1}{4} n s^2 \cot(\frac{\pi}{n})$
- Area (with apothem): $\frac{1}{2} \cdot n \cdot s \cdot a$

3D Shapes

- **Cube:** Volume $V = s^3$, Surface Area $SA = 6s^2$
- **Sphere:** Volume $V = \frac{4}{3} \pi r^3$, Surface Area $SA = 4\pi r^2$
- **Cylinder:** Volume $V = \pi r^2 h$, Surface Area $SA = 2\pi r^2 + 2\pi rh$
- **Cone:** Volume $V = \frac{1}{3} \pi r^2 h$, Surface Area $SA = \pi rs + \pi r^2$, where $s = \sqrt{h^2 + r^2}$
- **Cuboid:** Volume $V = lwh$, Surface Area $SA = 2(wh + lw + lh)$

Quadrilaterals

With side lengths a, b, c, d, diagonals e, f, diagonals angle θ, area A and magic flux $F = b^2 + d^2 - a^2 - c^2$:

$$4A = 2ef \cdot \sin\theta = F \tan\theta = \sqrt{4e^2 f^2 - F^2}$$

For cyclic quadrilaterals the sum of opposite angles is $180°$, $ef = ac + bd$, and $A = \sqrt{(p-a)(p-b)(p-c)(p-d)}$

Pick's Theorem

For a polygon on a grid:

$$A = I + \frac{B}{2} - 1$$

$A =$ Area, $I =$ Interior points, $B =$ Boundary points.

Spherical coordinates

$$x = r \sin\theta \cos\phi \qquad r = \sqrt{x^2 + y^2 + z^2}$$
$$y = r \sin\theta \sin\phi \qquad \theta = \mathrm{acos}(z/\sqrt{x^2 + y^2 + z^2})$$
$$z = r \cos\theta \qquad \phi = \mathrm{atan2}(y, x)$$

Coordinate Geometry

- **Distance (2 points):** $(x_1, y_1), (x_2, y_2)$ $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- **Midpoint:** $M = \left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$
- **Slope (2 points):** $m = \frac{y_2-y_1}{x_2-x_1}$
- **Line (point-slope):** $y - y_1 = m(x - x_1)$
- **Line (slope-intercept):** $y = mx + b$
- **Line (two-point):** $y - y_1 = \frac{y_2-y_1}{x_2-x_1}(x - x_1)$
- **Line (general):** $Ax + By + C = 0$
- **Slope (from general):** $m = -A/B$
- **Parallel lines:** have the same slope ($m_1 = m_2$)
- **Perpendicular lines:** $m_1 = -1/m_2$
- **Distance (point to line):** Point (x_0, y_0) to line $Ax+By+C = 0$. $D = \frac{|Ax_0+By_0+C|}{\sqrt{A^2+B^2}}$
- **Area of Triangle (vertices):** $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ $A = \frac{1}{2}|x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$
- **Circle:** Center (h, k), radius r. $(x - h)^2 + (y - k)^2 = r^2$
- **Distance (2 circle centers):** $D = \sqrt{(h_2 - h_1)^2 + (k_2 - k_1)^2}$
- **Tangent slope on circle:** At point (x_0, y_0) on circle $x^2+y^2 = r^2$. $m = -x_0/y_0$
- **Area of Parallelogram (vertices):** $(x_1, y_1), \ldots, (x_4, y_4)$ $A = |x_1y_2+x_2y_3+x_3y_4+x_4y_1-x_2y_1-x_3y_2-x_4y_3-x_1y_4|$
- **Ellipse:** $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$
- **Hyperbola:** $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$
- **Parabola:** Vertex (h, k), focus $(h + p, k)$. $(x - h) = 4p(y - k)$

Derivatives/Integrals

$$\frac{d}{dx}\arcsin x = \frac{1}{\sqrt{1-x^2}} \qquad \frac{d}{dx}\arccos x = -\frac{1}{\sqrt{1-x^2}}$$
$$\frac{d}{dx}\tan x = 1 + \tan^2 x \qquad \frac{d}{dx}\arctan x = \frac{1}{1+x^2}$$
$$\int \tan ax = -\frac{\ln|\cos ax|}{a} \qquad \int xe^{ax}dx = \frac{e^{ax}}{a^2}(ax - 1)$$
$$\int e^{-x^2} = \frac{\sqrt{\pi}}{2}\mathrm{erf}(x) \qquad \int x \sin ax = \frac{\sin ax - ax\cos ax}{a^2}$$

$$\int_a^b f(x)g(x)dx = [F(x)g(x)]_a^b - \int_a^b F(x)g'(x)dx$$

Sums

Basic Sums

- $\sum_{i=1}^n 1 = n$
- $\sum_{i=1}^n i = \frac{n(n+1)}{2}$
- $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{i=1}^n i^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^2(n+1)^2}{4}$
- $\sum_{i=1}^n i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$
- Sum of first n odd: $\sum_{i=1}^n (2i-1) = n^2$
- Sum of first n even: $\sum_{i=1}^n 2i = n(n+1)$

Arithmetic Progression (AP)

$a_n = a_1 + (n-1)d$ $S_n = \frac{n}{2}(2a_1 + (n-1)d) = \frac{n}{2}(a_1 + a_n)$ $a_n = a_m + (n-m)d$

Geometric Progression (GP)

$a_n = a_1 r^{(n-1)}$ $S_n = \frac{a_1(r^n-1)}{r-1}$ (finite) $S_\infty = \frac{a_1}{1-r}$ (for $|r| < 1$) $P_n = a_1^n r^{n(n-1)/2}$ $c^a + c^{a+1} + \cdots + c^b = \frac{c^{b+1}-c^a}{c-1}, c \neq 1$

Bernoulli Numbers & Sum of Powers

Pattern: Compute $\sum_{i=1}^n i^k$ where n **is large** but k **is small**.

- "Find $(1^5 + 2^5 + \cdots + n^5) \pmod{10^9 + 7}$ for $n = 10^{18}$."

Sequence B_k for $k = 0, 1, 2, \ldots$:

$$1, \frac{1}{2}, \frac{1}{6}, 0, -\frac{1}{30}, 0, \frac{1}{42}, 0, -\frac{1}{30}, \ldots$$

(Note: Using $B_1 = +1/2$. The $B_1 = -1/2$ convention also exists.)

EGF for B_k (using $B_1 = -1/2$):

$$\frac{x}{e^x - 1} = \sum_{k=0}^{\infty} B_k \frac{x^k}{k!}$$

Faulhaber's Formula (Sum of Powers):

$$\sum_{i=0}^{n-1} i^m = \frac{1}{m+1}\sum_{k=0}^{m}\binom{m+1}{k}B_k n^{m+1-k}$$

Combinatorics

Binomial Theorem

Description: Used for expanding powers of binomials $(a + b)^p$. The coefficients $\binom{p}{k}$ give the number of ways to choose k items from p.
Formula:

$$(a+b)^p = \sum_{k=0}^{p}\binom{p}{k}a^k b^{p-k}$$

Stars and Bars

Description: Used to find the number of ways to distribute **identical (unlabeled)** objects (n) into **distinct** bins (k).
Formulas:

- **Empty bins NOT valid (Positive Integer Solutions):** $\binom{n-1}{k-1}$
- **Empty bins VALID (Non-Negative Integer Solutions):** $\binom{n+k-1}{k-1}$

Binomial Coefficients $\binom{n}{k}$

Description: $\binom{n}{k}$ is the number of ways to choose k elements from n distinct elements. Essential for DP, probability, and modular arithmetic.

- **Definition:** $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
- **Symmetry:** $\binom{n}{k} = \binom{n}{n-k}$
- **Multiplicative ($\mathcal{O}(k)$):** $\binom{n}{k} = \prod_{i=1}^k \frac{n-i+1}{i}$
- **Base Cases:** $\binom{n}{0} = 1$, $\binom{n}{n} = 1$
- **Pascal's Identity ($\mathcal{O}(1)$ DP):** $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$
- **Absorption Identity:** $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$
- **Shifted Recurrence I:** $\binom{n}{k} = \frac{n-k+1}{k}\binom{n}{k-1}$
- **Shifted Recurrence II:** $\binom{n+1}{k} = \frac{n+1}{n-k+1}\binom{n}{k}$
- **Vandermonde's Identity:** $\binom{m+n}{r} = \sum_{k=0}^r \binom{m}{k}\binom{n}{r-k}$
- **Hockey-Stick Identity:** $\sum_{i=k}^n \binom{i}{k} = \binom{n+1}{k+1}$
- **Sum of Row (Total Subsets):** $\sum_{k=0}^n \binom{n}{k} = 2^n$
- **Sum of K (Weighted Sum):** $\sum_{k=1}^n k\binom{n}{k} = n2^{n-1}$
- **Sum of K^2 (Weighted Sum II):** $\sum_{k=1}^n k^2\binom{n}{k} = n(n+1)2^{n-2}$

Stirling Numbers of the First Kind: $c(n, k)$

Pattern: Count permutations in terms of their **cycle structure**.

- "Arrange n people around k identical round tables."
- "Count permutations of n elements with exactly k cycles."

Definition: Number of permutations of n items with k cycles.

$$c(n,k) = (n-1)c(n-1,k) + c(n-1,k-1)$$
$$c(n,0) = 0 \quad (n > 0), \quad c(0,0) = 1$$

$$\sum_{k=0}^{n} c(n,k)x^k = x(x+1)\ldots(x+n-1)$$

Sequence $c(n, 2)$ for $n = 0, 1, 2, \ldots$:

$$0, 0, 1, 3, 11, 50, 274, 1764, 13068, \ldots$$

Stirling Numbers of the Second Kind: $S(n, k)$ or $\left\{{n \atop k}\right\}$

Pattern: Partition n **distinct items** into k **identical, non-empty boxes**.

- "How many ways to put n *labeled* balls into k *unlabeled* boxes?"
- "Count ways to partition a set of n elements into k non-empty subsets."

Definition: Number of partitions of n distinct elements into exactly k non-empty subsets.

$$S(n,k) = S(n-1,k-1) + k \cdot S(n-1,k)$$
$$S(n,1) = 1, \quad S(n,n) = 1$$
$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n$$

Bell Numbers: $B(n)$

Pattern: Total ways to partition n **distinct items** (number of boxes doesn't matter).

- "Find the total number of equivalence relations on a set of n elements."
- "How many ways to put n *labeled* balls into *unlabeled* boxes?"

Definition: Total number of partitions of n distinct elements.

$$B(n) = \sum_{k=0}^{n} S(n,k)$$

Sequence $B(n)$ for $n = 0, 1, 2, \ldots$:

$$1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, \ldots$$

Recurrence (P-set construction):

$$B(n+1) = \sum_{k=0}^{n}\binom{n}{k}B(k)$$

Catalan Numbers: C_n

Pattern: One of the most famous sequences. Look for:

- **Balanced sequences:** "Valid (balanced) parenthesis strings of length $2n$."
- **Recursive splitting:** "Number of full binary trees with n nodes."
- **Non-crossing paths:** "Paths from (0, 0) to (n, n) on a grid that do not go above $y = x$."
- **Polygon triangulation:** "Ways to triangulate a convex polygon with $n+2$ sides."

Sequence C_n for $n = 0, 1, 2, \ldots$:

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, \ldots$$

Closed Form:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \binom{2n}{n} - \binom{2n}{n+1}$$

Recurrence Relations:

$$C_0 = 1, \quad C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$$

$$C_0 = 1, \quad C_{n+1} = \frac{2(2n+1)}{n+2} C_n$$

Eulerian Numbers: $E(n,k)$

Pattern: Count permutations based on their **"runs"** or **"ascents/descents"**.

- "Count permutations of $\{1, \ldots, n\}$ with exactly k ascents ($p_i < p_{i+1}$)."

Definition: Number of n-permutations with exactly k rises (positions i with $p_i > p_{i-1}$).

$$E(n,k) = (n-k)E(n-1,k-1) + (k+1)E(n-1,k)$$
$$E(n,0) = E(n,n-1) = 1$$
$$E(n,k) = \sum_{j=0}^{k} (-1)^j \binom{n+1}{j} (k-j+1)^n$$

Derangements: $D(n)$ or $!n$

Pattern: The "mixed-up hats" or "secret santa" problem.

- "Count permutations of n elements where **no element is in its original position**."
- "Find the number of permutations with **no fixed points** ($p_i \neq i$ for all i)."

Definition: Permutations of a set such that no element appears in its original position. **Sequence $D(n)$ for $n = 0, 1, 2, \ldots$:**

$$1, 0, 1, 2, 9, 44, 265, 1854, 14833, \ldots$$

Recurrence Relations:

$$D(n) = (n-1)(D(n-1) + D(n-2))$$
$$D(n) = n \cdot D(n-1) + (-1)^n$$
$$D(n) = \left\lfloor \frac{n!}{e} + \frac{1}{2} \right\rfloor = \left\lceil \frac{n!}{e} \right\rceil \quad (n \geq 1)$$

Burnside's Lemma

Pattern: Count "distinct" objects under **symmetry** (rotations, reflections).

- "Count distinct ways to color a necklace/bracelet/cube under rotation."
- The key is "up to symmetry," "distinct under rotation," etc.

Definition: Given a group G of symmetries acting on a set X. The number of distinct elements of X up to symmetry (number of orbits) is:

$$|X/G| = \frac{1}{|G|} \sum_{g \in G} |X^g|$$

where $X^g = \{x \in X \mid g \cdot x = x\}$ are the elements fixed by g.

Special Case (Necklaces): For k colors and n beads, with $G = \mathbb{Z}_n$ (rotations):

$$\text{Count} = \frac{1}{n} \sum_{d \mid n} \phi(d) \cdot k^{n/d}$$

Permutation Cycles (EGF)

Pattern: Count permutations where **cycle lengths are restricted** to a set S.

- "Count permutations of n elements that consist *only* of cycles of length 2 (involutions)."

Definition: Let $g_S(n)$ be the number of n-permutations whose cycle lengths all belong to S. The Exponential Generating Function (EGF) is:

$$\sum_{n \geq 0} g_S(n) \frac{x^n}{n!} = \exp\left(\sum_{n \in S} \frac{x^n}{n} \right)$$

Lucas's Theorem

Pattern: Compute $\binom{n}{k} \pmod p$ where n, k **are large** but p **is a small prime**.

- "Calculate $\binom{10^{18}}{10^9} \pmod 7$."

Definition: Let n, m be non-negative integers and p a prime. Write n and m in base p:

$$n = n_k p^k + \cdots + n_1 p + n_0$$
$$m = m_k p^k + \cdots + m_1 p + m_0$$

Then:

$$\binom{n}{m} \equiv \prod_{i=0}^{k} \binom{n_i}{m_i} \pmod p$$

(Note: $\binom{a}{b} = 0$ if $a < b$)

Series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots, \; (-\infty < x < \infty)$$
$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots, \; (-1 < x \leq 1)$$
$$\sqrt{1+x} = 1 + \frac{x}{2} - \frac{x^2}{8} + \frac{2x^3}{32} - \frac{5x^4}{128} + \ldots, \; (-1 \leq x \leq 1)$$
$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots, \; (-\infty < x < \infty)$$
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots, \; (-\infty < x < \infty)$$
$$(1-x)^{-r} = \sum_{i=0}^{\infty} \binom{r+i-1}{i} x^i, \; (r \in \mathbb{R})$$

Bitwise Formulas

$$a|b = a \oplus b + a\&b$$
$$a \oplus (a\&b) = (a|b) \oplus b \qquad a \oplus b = (a\&b) \oplus (a|b)$$
$$a + b = a|b + a\&b \qquad a + b = a \oplus b + 2(a\&b)$$

$a - b = (a \oplus (a\&b)) - ((a|b) \oplus a) = ((a|b) \oplus b) - ((a|b) \oplus a) = (a \oplus (a\&b)) - (b \oplus (a\&b)) = ((a|b) \oplus b) - (b \oplus (a\&b))$

Algorithms

Rotation of a n*m matrix: $(i,j) \to (j, n-i-1) \to (n-i-1, m-j-1) \to (m-j-1, i)$

Probability theory

Let X be a discrete random variable with probability $p_X(x)$ of assuming the value x. It will then have an expected value (mean) $\mu = \mathbb{E}(X) = \sum_x x p_X(x)$ and variance $\sigma^2 = V(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2 = \sum_x (x - \mathbb{E}(X))^2 p_X(x)$ where σ is the standard deviation. If X is instead continuous it will have a probability density function $f_X(x)$ and the sums above will instead be integrals with $p_X(x)$ replaced by $f_X(x)$.

Expectation is linear:

$$\mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y)$$

For independent X and Y,

$$V(aX + bY) = a^2 V(X) + b^2 V(Y).$$

Discrete distributions

Binomial distribution: The number of successes in n independent yes/no experiments, each which yields success with probability p is $\mathrm{Bin}(n,p)$, $n = 1, 2, \ldots, 0 \leq p \leq 1$.

$$p(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$$\mu = np, \; \sigma^2 = np(1-p)$$

$\mathrm{Bin}(n,p)$ is approximately $\mathrm{Po}(np)$ for small p.

First success distribution: The number of trials needed to get the first success in independent yes/no experiments, each which yields success with probability p is $\mathrm{Fs}(p)$, $0 \leq p \leq 1$.

$$p(k) = p(1-p)^{k-1}, \; k = 1, 2, \ldots$$

$$\mu = \frac{1}{p}, \; \sigma^2 = \frac{1-p}{p^2}$$

Poisson distribution: The number of events occurring in a fixed period of time t if these events occur with a known average rate κ and independently of the time since the last event is $\mathrm{Po}(\lambda)$, $\lambda = t\kappa$.

$$p(k) = e^{-\lambda} \frac{\lambda^k}{k!}, k = 0, 1, 2, \ldots$$

$$\mu = \lambda, \; \sigma^2 = \lambda$$

Continuous distributions

Uniform distribution: If the probability density function is constant between a and b and 0 elsewhere it is $\mathrm{U}(a,b)$, $a < b$.

$$f(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & \text{otherwise} \end{cases}$$

$$\mu = \frac{a+b}{2}, \; \sigma^2 = \frac{(b-a)^2}{12}$$

Exponential distribution: The time between events in a Poisson process is $\mathrm{Exp}(\lambda)$, $\lambda > 0$.

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$\mu = \frac{1}{\lambda}, \; \sigma^2 = \frac{1}{\lambda^2}$$

Normal distribution: Most real random values with mean μ and variance σ^2 are well described by $\mathcal{N}(\mu, \sigma^2)$, $\sigma > 0$.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

If $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ then

$$aX_1 + bX_2 + c \sim \mathcal{N}(\mu_1 + \mu_2 + c, a^2\sigma_1^2 + b^2\sigma_2^2)$$

Graph Theory

Cayley's Formula

Pattern: Count **spanning trees** on n **labeled** vertices in a **complete graph** K_n.

- "How many trees can be formed using n labeled nodes?"

Definition: The number of spanning trees on n labeled vertices (in K_n) is n^{n-2}. **Sequence** n^{n-2} **for** $n = 1, 2, 3, \ldots$**:**

$$1, 1, 3, 16, 125, 1296, 16807, \ldots$$

Generalizations:

- # with degrees d_i: $\frac{(n-2)!}{(d_1-1)!(d_2-1)!\ldots(d_n-1)!}$ (Prufer Sequence)

Kirchhoff's Matrix Tree Theorem

Pattern: Count **spanning trees** in a **general graph** G (not complete).

- "Given a grid graph, find the number of spanning trees."

Definition: Counts spanning trees in a graph G.

1. Create the **Laplacian Matrix** $L = D - A$:
   - D = Degree Matrix (diagonal, $D_{ii} = \deg(i)$)
   - A = Adjacency Matrix

   Or, $L_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$
2. Remove **any** row i and **any** column j to get $L_{i,j}$.
3. The number of spanning trees is $\det(L_{i,j})$.

Erdős–Gallai Theorem

Pattern: Given a sequence of numbers, can it be the **degree sequence** of a **simple graph**?

- "Is the sequence $d_1, \ldots, d_n$ a valid graphic sequence?"

Definition: A simple graph with node degrees $d_1 \geq \cdots \geq d_n$ exists iff:

1. $\sum_{i=1}^{n} d_i$ is even.
2. For every $k \in [1, n]$:

$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min(d_i, k)$$

Game Theory

Sprague–Grundy Theorem

Pattern: An **impartial game** (moves depend on position, not player).

- **Classic Nim:** "A game with multiple piles of stones."
- **Sum of games:** Game breaks into independent sub-games.

Definition: For impartial games.

- **Grundy Value (G-value) / Nim-sum:**

$$G(v) = \text{mex}(\{G(v_i) \mid v \to v_i \text{ is a valid move}\})$$

where $\text{mex}(S)$ is the Minimum Excluded value.
- **Losing Position:** $G(v) = 0$.
- **Winning Position:** $G(v) > 0$.
- **Sum of Games:** If a game is a sum of independent games $g_1, \ldots, g_k$:

$$G_{\text{total}} = G(g_1) \oplus G(g_2) \oplus \cdots \oplus G(g_k)$$

where $\oplus$ is the bitwise XOR operator.

Trivia

Pythagorean triples: The Pythagorean triples are uniquely generated by $a = k\cdot(m^2 - n^2)$, $b = k\cdot(2mn)$, $c = k\cdot(m^2 + n^2)$ with $m > n > 0$, $k > 0$, $\gcd(m, n) = 1$, both m, n not odd.
Primes: $p = 962592769$ is such that $2^{21} \mid p - 1$, which may be useful. For hashing use 970592641 (31-bit number), 31443539979727 (45-bit), 3006703054056749 (52-bit). There are 78498 primes less than 1 000 000.
Estimates: $\sum_{d|n} d = O(n \log \log n)$.
Prime Gaps: For primes $> 10^{12}$, the max gap is not definitively known, but a gap of 1600 is a safe upper bound for practical purposes. (The largest known gap is 1550).
Prime count: 5133 upto 5e4. 9592 upto 1e5. 17984 upto 2e5. 78498 upto 1e6. 5761455 upto 1e8.
max NOD $\leq n$: 100 for $n = 5e4$. 500 for $n = 1e7$. 2000 for $n = 1e10$. 200 000 for $n = 1e19$.
max Unique Prime Factors: 6 upto 5e5. 7 upto 9e6. 8 upto 2e8. 9 upto 6e9. 11 upto 7e12. 15 upto 3e19.
Quadratic Residue: $(\frac{a}{p})$ is 0 if $p|a$, 1 if a is a quadratic residue, -1 otherwise. Euler: $(\frac{a}{p}) = a^{(p-1)/2} \pmod p$ (prime). Jacobi: if $n = p_1^{e_1} \cdots p_k^{e_k}$ then $(\frac{a}{n}) = \prod (\frac{a}{p_i})^{e_i}$.
Chicken McNugget: If a, b coprime, there are $\frac{1}{2}(a-1)(b-1)$ numbers not of form $ax + by$ $(x, y \geq 0)$, the largest being $ab - a - b$.

Template & Utils

PBDS (Ordered Set & Hash Map)

Description: 1. `orderS`: RB-Tree. Supports `find_by_order(k)` (k-th element) and `order_of_key(x)` (count strictly less than x). 2. `hash_map`: Faster than `std::unordered_map`. Uses `custom_hash` to prevent anti-hash tests.

```
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;
using orderS = tree<ll,null_type,less<ll>,
    ↪ rb_tree_tag,
    ↪ tree_order_statistics_node_update>;

struct custom_hash {
    static uint64_t splitmix64(uint64_t x)
        ↪ {
        x += 0x9e3779b97f4a7c15;
        x = (x ^ (x >> 30)) * 0
            ↪ xbf58476d1ce4e5b9;
        x = (x ^ (x >> 27)) * 0
            ↪ x94d049bb133111eb;
        return x ^ (x >> 31);
    }
    size_t operator()(uint64_t x) const {
        static const uint64_t FIXED_RANDOM
            ↪ =
            chrono::steady_clock::now().
                ↪ time_since_epoch().
                ↪ count();
        return splitmix64(x + FIXED_RANDOM)
            ↪ ;
    }
};
template <typename K, typename V>
using hash_map = gp_hash_table<K, V,
    ↪ custom_hash>;
```

Pragmas & Optimization

Description: Aggressive GCC optimizations. `Ofast` ignores strict IEEE floating point standards (be careful with geometry precision).

```
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast,unroll-loops")
#pragma GCC optimize("tree-vectorize")
#pragma GCC target("avx2,sse4.2,popcnt")
```

Random Number Generator

Description: Mersenne Twister (`mt19937`) seeded with high-resolution clock. Much better than `rand()`.

```
mt19937 rng(chrono::high_resolution_clock::
    ↪ now().time_since_epoch().count());
inline ll getrandom(ll a,ll b) { return
    ↪ uniform_int_distribution<ll>(a,b)(
    ↪ rng); }
```

Basic Math Utils

Description: 1. `bigmod`: Modular Exponentiation $\mathcal{O}(\log P)$. 2. `inversemod`: Modular Inverse using Fermat's Little Theorem (Requires Prime Mod). 3. `sqrtt`: Integer Square Root (avoids precision errors of `sqrt`).

```
ll bigmod(ll base, ll power) {
    ll res = 1; ll p = base % mod;
    while (power > 0) {
        if (power % 2 == 1) res = ((res %
            ↪ mod) * (p % mod)) % mod;
        power /= 2;
        p = ((p % mod) * (p % mod)) % mod;
    }
    return res;
}
ll inversemod(ll base) { return bigmod(base
    ↪ , mod - 2); }

int gcd(ll a, ll b) {
    while (b) { a %= b; swap(a, b); }
    return a;
}
ll sqrtt(ll a) {
    long long x = sqrt(a) + 2;
    while (x * x > a) x--;
    return x;
}
```

Grid Moves (2D)

Description: Direction arrays for implicit graphs (grids).

```
// 4 Directions: Up, Down, Left, Right
// 8 Directions: Adds Diagonals
int dx[]={-1, 1 , 0 , 0 , -1 ,-1, 1, 1};
int dy[]={ 0, 0 ,-1 , 1 , -1 , 1,-1, 1};
// up = {-1,0}, down = {1,0}, right =
    ↪ {0,1}, left = {0,-1}
constexpr ld PI =
    ↪ 3.14159265358979323846264338327950288
    ↪ L;
```

Fast I/O & Debug

```
// Place inside main
ios::sync_with_stdio(0); cin.tie(0);
cout.setf(ios::fixed); cout.precision(10);

// File I/O helper
inline void file() {
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
}
// Timer
clock_t start= clock();
cerr << "Time: " <<((double)(clock() -
    ↪ start) / CLOCKS_PER_SEC)<<el;
```

CP Environment Setup

Setup Procedure

Description: Guide to establishing the fast-testing environment in a Linux terminal.

1. **Transfer Files:** Create the necessary files (`stdc.h`, `template.cpp`, `cf`, `rte`, `stress`, `gen.cpp`, `right.cpp`, `wrong.cpp`) and paste the content below.
2. **Permissions:** In the terminal, run: `chmod +x cf rte stress gen.cpp`.
3. **Execution Alias:** Run the following alias commands **once per session** to enable short commands like `cf A.cpp`:

```
alias cf='./cf '
alias rte='./rte '
alias stress='./stress '
```

4. **Workflow:** To test `A.cpp`, type `cf A.cpp`. To debug, type `rte A.cpp`. To stress test, type `stress`.

C++ Library Header (stdc.h)

Description: Contains all necessary includes, complex debugging macros, and types. Run `g++ stdc.h -o stdc.h.gch` once to enable fast precompilation.

```cpp
#include <bits/stdc++.h>
using namespace std;
#define TT template <typename T>

// --- Complex Variadic Template Debugger
    ↪ Logic (Preserved) ---

TT, typename=void> struct cerr_ok :
    ↪ false_type {};
TT> struct cerr_ok<T, void_t<decltype(cerr
    ↪ << declval<T>())>> : true_type {};

TT> constexpr void p1(const T &x);

TT, typename V> void p1(const pair<T, V> &x
    ↪ )
{
    cerr << "{";
    p1(x.first);
    cerr << ", ";
    p1(x.second);
    cerr << "}";
}
TT> constexpr void p1(const T &x)
{
    if constexpr (cerr_ok<T>::value) cerr
        ↪ << x;
    else
    {
        int f = 0;
        cerr << "{";
        for (auto &i : x)
            cerr << (f++ ? "," : ""), p1(i)
                ↪ ;
        cerr << "}";
    }
}
void p2() { cerr << "]\n"; }
TT, typename... V> void p2(T t, V... v)
{
    p1(t);
    if (sizeof...(v))
        cerr << ", ";
    p2(v...);
}
#ifdef DeBuG
#define dbg(x...) { cerr << "\t\e[93m" << \
    __func__ << ":" << __LINE__ << "[" << #
        ↪ x << "] " << \
    "= ["; p2(x); cerr << "\e[0m"; }
#endif
```

Base Solution File (template.cpp)

Description: The starting file for every problem. Includes the necessary macros and I/O setup.

```cpp
// IIUC_MARK_US
#include "bits/stdc++.h"
using namespace std;

#ifndef DeBuG
#define dbg(...)
#endif

#define sz(x) (int) (x).size()
#define all(x) begin(x), end(x)
#define rep(i, a, b) for (int i = a; i < (b
    ↪ ); ++i)
using ll = long long; using pii = pair<int,
    ↪ int>;
using pll = pair<ll, ll>; using vi = vector
    ↪ <int>;
template<class T> using V = vector<T>;

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

}
```

Fast Compile & Run (cf)

Description: Compiles the specified file (e.g., `A.cpp`) with full optimization (`-O2`) and runs it.

```bash
#!/bin/bash
TARGET_FILE=$1

if [ -z "$TARGET_FILE" ]; then
    echo "Usage: cf <source_file.cpp>"
    exit 1
fi

g++ -o sol -Wall -Wextra -std=c++17 -O2 "
    ↪ $TARGET_FILE"

if [ $? -eq 0 ]; then
    echo "--- Running $TARGET_FILE ---"
    time ./sol < input.txt
fi
```

Debug Check (rte)

Description: Compiles with Sanitizers to catch memory/integer overflow errors (ASAN, UBSAN).

```bash
#!/bin/bash
TARGET_FILE=$1

if [ -z "$TARGET_FILE" ]; then
    echo "Usage: rte <source_file.cpp>"
    exit 1
fi
# Compile with Sanitizers (Address and
    ↪ Undefined Behavior)
g++ -o sol -std=c++17 -O2 -Wall -Wextra \
-fsanitize=address,undefined "$TARGET_FILE"

if [ $? -eq 0 ]; then
    echo "--- Running sanitizer check on
        ↪ $TARGET_FILE ---"
    ./sol < input.txt
fi
```

Stress Test Script (stress)

Description: Continuous verification tool. Compiles `right.cpp` (verified solution) and `wrong.cpp` (optimized solution) and tests them against random inputs from `./gen`.

```bash
#!/bin/bash
# TARGET FILES: right.cpp (verified) and
    ↪ wrong.cpp (tested/optimized)

# 0. Compile the test case generator
g++ -o gen gen.cpp -std=c++17 -O2

# 1. Compile the verified solution (RIGHT
    ↪ ANSWER)
g++ -o right right.cpp -std=c++17 -Wall -O0
    ↪ -D_GLIBCXX_DEBUG

# 2. Compile the optimized solution (
    ↪ POTENTIALLY WRONG ANSWER)
g++ -o wrong wrong.cpp -std=c++17 -Wall -O2

for ((i = 1; ; ++i)); do
    echo "Testing case $i..."

    # Generate input using the C++
        ↪ executable, passing the case
        ↪ number $i$ as the seed
    ./gen $i > input.txt

    # Run solutions and capture output
    ./right < input.txt > right.out
    ./wrong < input.txt > wrong.out

    # Check if outputs differ
    if ! diff -w right.out wrong.out; then
        echo "--- Found Difference! Failing
            ↪ Case Saved to input.txt
            ↪ ---"
        break
    fi
done
```