Solutions to selected problems on Timus online judge

Wang Xiang

January 4, 2023

Contents

Problem 1 (1058 Chocolate)**.** Key point is to calculate the minimal length of pq for four points A, B, C, D satisfying that $S_{\triangle AXB} = S_{\triangle CXD}$ where $X = AD \cap BC$, $p \in AB, q \in CD$ and $S_{\triangle AYp} = S_{\triangle DYq}$. $AC//BD$ since $S_{\triangle AXB} = S_{\triangle CXD}$. Let $Z = AB \cap CD$, if $(\cos ZAD - \cos ZDA) * (\cos ZBC - \cos ZCB) < 0$, then the length of pq is given by

$$(\frac{pq}{2})^2 = S_{\triangle ZAD} \tan \frac{Z}{2} = \frac{\lambda}{1-\lambda} S_{\triangle BAD} \frac{\sin Z}{1+\cos Z},$$

$$\lambda = \frac{AC}{BD}, \quad \sin Z = \frac{AB \times CD}{|AB||CD|}, \quad \cos Z = \frac{AB \cdot CD}{|AB||CD|},$$

$$AB \times CD = AB \times D'A = BD' \times BA = (1-\lambda) BD \times BA, \quad S_{\triangle BAD} = \frac{AB \times AD}{2},$$

$$pq = \sqrt{\frac{2\lambda (AB \times AD)^2}{|AB||CD|(1+\cos Z)}}$$

Problem 2 (1132 Square Root)**.** Find the square root of a modulo a prime n.

Problem 3 (1172 Ship Routes)**.** Dynamic programming, calculate the number of strings s using N characters of A, B, C each, such that $s[0] == A$ and no 2 consecutive characters are identical. Let $dp[a, b, c, ch]$ be the number of strings starting with A, numbers of A, B, C are a, b, c respectively, and that ends with $ch \in \{A, B, C\}$. The answer is

$$\frac{(dp[N, N, N, B] + dp[N, N, N, C])(N-1)!N!^2}{2},$$

Problem 4 (1199 Mouse)**.** Single source shortest path using Dijkstra. Key point is to generate the path of the mouse. Use $i = (i + N - 1)\%N$ instead of $i = (i - 1)\%N$ since taking modulo on negative integers will produce negative answers.

Problem 5 (1239 Ghost Busters)**.** Project the ghosts onto the unit sphere, they become spherical circles. Preprocess the ghosts so that their center lie on the unit sphere, we may assume ghost i has center $po_i = (x_i, y_i, z_i)$ and radius r_i. An spherical circle has a plane it lies on and radius in radian:

$$xx_i + yy_i + zz_i = c_i = \sqrt{1 - r_i^2}, \quad rad = \arcsin r_i,$$

For circle i and j, their intersections are determined as follows:

$$A = \begin{pmatrix} 1 & po_i \cdot po_j \\ po_j \cdot po_i & 1 \end{pmatrix}, \quad \begin{pmatrix} u \\ v \end{pmatrix} = A^{-1} \begin{pmatrix} c_i \\ c_j \end{pmatrix}, \quad \begin{pmatrix} x_{mid} \\ y_{mid} \\ z_{mid} \end{pmatrix} = \begin{pmatrix} x_i & x_j \\ y_i & y_j \\ z_i & z_j \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}$$

Here p_{mid} is the projection of the origin onto the line intersection of planes i and j. The above calculation find the least square solution to the equation below

$$xx_i + yy_i + zz_i = c_i, \quad xx_j + yy_j + zz_j = c_j,$$

The direction of the line is $dir = \frac{po_i \times po_j}{|po_i \times po_j|}$, with half the segment length $halfseg = \sqrt{1 - |p_{mid}|^2}$. So the intersections are

$$I_1 = p_{mid} + dir * halfseg, \quad I_2 = p_{mid} - dir * halfseg,$$

Scan the $\frac{1}{8}$ unit sphere from the north pole to equator. There are only three cases that would change the order of intersection segments of circles: inserting north endpoint, deleting south endpoint, and intersection between circles. So my strategy is to enumerate all such critical latitudes, for each latitude, scan from longitude 0 to $\frac{\pi}{2}$ and record the location that meets the most segments. The original formula of inverse of a $2 * 2$ matrix has a bug, but now it is fixed.

Problem 6 (1266 Kirchhoff's Law)**.** Rewrite Solve() function to remove L matrix, and updated all occurences of this function. Wrote *findstringinfiles.sh* to find all occurences of Solve() function. Express the problem as a linear system with n variables and n equations, variables are potentials at each node. At node $0, n-1$, the two equations are

$$potential[0] = 1, \quad potential[n-1] = 0,$$

At node i where $1 \leq i \leq n-2$, the equations are

$$\sum_{j<i} (potential[j] - potential[i]) rinv[j][i] = \sum_{i<j} (potential[i] - potential[j]) rinv[i][j],$$

where $rinv[i][j]$ is the inverse resistence from node i to node j, $i < j$.

Problem 7 (1286 Starship Travel)**.** We consider the arithmetic of the ring of Gaussian integers $\mathbb{Z}[i]$, which is an Euclidean domain.

$$\alpha = p + qi, \quad \overline{\alpha} = p - qi, \quad i\overline{\alpha} = q + pi, \quad d = \gcd(\alpha, \overline{\alpha}),$$

Assume that the initial and final positions are $b, c \in \mathbb{Z}[i]$, then the starship can move to the destination if and only if $d \mid c - b$. Notice that $\mathbb{Z}[i]$ is a principal ideal domain, so we have $(d) = (\alpha, \overline{\alpha})$.

Problem 8 (1310 ACM Diagnostics)**.** Method 1: raw calculation and it received TLE-16. Method 2: dynamic programming, let $dp[i][j]$ be the number of i-digits states that their sum equals j modulo K. Initially $dp[0][0] = 1$, state transition equation is $dp[i][j] += dp[i-1][l] * num$, num is the number of integers in $[1, M]$ that equals $j - l$ modulo K. Then the answer state is calculated from the highest digit to the lowest digit.

Problem 9 (1318 Logarithm). Bitwise not operator is $\sim$ in c++.

$$\sum_{j,k} \lfloor \log_{10}(a_j \wedge a_k) \rfloor = \sum_{i=1}^{38} \sum_{j,k} (a_j \wedge a_k) \geq 10^i,$$

Another approach is to split $[0, 10^i - 1], 1 \leq i \leq 38$ into 869 intervals.

$$\overline{I} = [0, 10^i - 1], \quad \overline{I_k} = \overline{I} \wedge a_k,$$

$\overline{I}$ can be written as union of some intervals, so is $\overline{I_k}$.

Problem 10 (1368 Goat in the Garden 3). When $K = n^2 + (n+1)^2$, $ans = 4n+4$; when $K = 2n^2$, $ans = 4n+2$; when $K = n(2n+1)$, $ans = 4n+3$; when $K = (2n+1)(n+1)$, $ans = 4n+5$. In the above four cases, we say that K is saturated. For non-saturated K value, construct the output of saturated K value first and modify it.

Problem 11 (1371 Cargo Agency). Centroid decomposition. Notice that I wrote a hash function to store the cost of an edge in it.

Problem 12 (1372 Death Star).

Problem 13 (1375 Bill Clevers). Given prime p and integer k, find x, y such that $x^2 + y^2 \equiv k \mod p$. Store $rootmap[i * i\%p] = i$ and query $rootmap[(k - i * i)\%p]$.

Problem 14 (1384 Goat in the Garden 4). Non-convex optimization. $dirnum = 40, stepsizenum = 18, stepsize = 16./(1 << i), 0 \leq i \leq 17$. Initial seeds are mid points of edges and polygon vertices. Actually I implemented a gradient descent algorithm adopted from Boyd's book "Convex Optimization". Initial directions are randomly selected before each step.

Problem 15 (1387 Vasya's Dad). Partition of an integer. Method 1: calculate all the partition of integer $N-1$ recursively.

$$dp[N] = \sum_{\sum_{i=1}^{l} n_i p_i = N-1} \prod_{i=1}^{l} \binom{dp[p_i] + n_i - 1}{n_i}, \quad p_1 > p_2 ... > p_l,$$

Method 2: memorized search. Use $memorizedstates$ dictionary to store calculated results of $solve(n, m)$. $solve(n, m)$ represents the number of different forests with n vertices in total and the size of trees not exceeding m. m is surely to decrease in recursion, and we calculate the multiplicity $\binom{dp[k]+m-1}{m}$ indicating m subtrees each size equal to k in each step.

Problem 16 (1396 Maximum Version 2). Suppose i is a maximum index, then i is odd, $i = 2i_1 + 1, i_2 = i_1 + 1$, then either 1) $i1$ is even, at least one of $\frac{i_1}{2}, i_2$ is a maximum index; or 2) i_2 is even, at least one of $\frac{i_2}{2}, i1$ is a maximum index. But I don't know how to prove it. Thus candidate indices from 2^n to 2^{n+1} can be generated from calculated maximum indices from 2^{n-2} to 2^n, in the sense that if i is a maximum index, then there exists an maximum index j such that at least one of the following four equalities holds

$$i = 4j + 1, \quad i = 4j - 1, \quad i = 2j + 1, \quad i = 2j - 1,$$

Problem 17 (1420 Integer-Valued Complex Division). Implemented struct GaussianQT in this problem. Since the norms of numerator and denominator of $\frac{a}{b}$ exceed the range of long long, I wrote BigInteger struct and got accepted for the first time.

Problem 18 (1421 Credit Operations)**.** Given two arrays indicating demands and bandwidths of vertices of a complete bipartite graph $K_{N,N}$, its edges' capacities being 100, determine whether the maximum flow is fully-loaded. Accepted using dinic maxflow algorithm copied from online implementation. Reference: *https://cp-algorithms.com/graph/dinic.html*.

Problem 19 (1430 Crime and Punishment)**.** Given $1 \le A, B, N \le 2e9$, find $x, y \ge 0$, such that $Ax + By \le N$, and $Ax + By$ attains maximum. Without loss of generality, assume that $A > B, \gcd(A, B) = 1$. If $B = 1$, then we can return the answer $0, N$ directly. Otherwise, let

$$A = kB + A_1, \quad k = \lfloor \frac{A}{B} \rfloor, \quad x_0 = \lfloor \frac{N}{A} \rfloor, \quad y_0 = \lfloor \frac{N - Ax_0}{B} \rfloor,$$

$$y_1 = y - y_0 - k(x_0 - x), \quad x_1 = x,$$

$$A_1 x_1 + By_1 = Ax + By - kBx_0 - By_0 \le N - kBx_0 - By_0 \triangleq N_1,$$

After translation and slope modification, exactly one of the following cases happens: 1) $x_0 = \lfloor \frac{N_1}{A_1} \rfloor$, then we can directly solve the case $A_1 x_1 + By_1 \le N_1$, and let $x = x_1, y = y_1 + y_0 + k(x_0 - x_1)$; 2) $x > \lfloor \frac{N_1}{A_1} \rfloor$ and $B > N_1$, then we must have $x_1 = x_0, y_1 = 0$, the formula to get x, y is identical to that in case 1); 3) $x_0 < \lfloor \frac{N_1}{A_1} \rfloor$ and $B \le N_1$, then we may discard those decision points that $y_1 = 0$, let $x_2 = x_1, y_2 = y_1 - 1$, then we have $A_1 x_2 + By_2 \le N_1 - B \triangleq N_2$, and we may solve the above case recursively.

Problem 20 (1449 Credit Operations 2)**.** Maximal weighted bipartite matching. Given a integer coefficient matrix $\{w_{i,j}\}, 1 \le i, j \le N$, find $v_i^l, v_j^r, 1 \le i, j \le N$, such that $v_i^l + v_j^r \ge w_{i,j}$ holds for every pair of i, j, and $\sum_{i=1}^N v_i^l + \sum_{j=1}^N v_j^r$ attains its minimum. Its dual is: find a perfect matching $M \in K_{N,N}$ such that the total weight of the matching $\sum_{(i,j)\in M} w_{i,j}$ attains its maximum. I implemented Hungarian algorithm, reference: *ti.inf.ethz.ch/ew/lehre/GT03/lectures/PDF/lecture6f.pdf*. This implementation can be viewed as a special case of primal-dual optimization method, because it solves primal and dual problems alternately.

Primal formulation: maximum weighted bipartite matching. $0 \le e_{i,j} \le 1$, $\sum_{1\le j\le N} e_{i,j} = 1$, $\sum_{1\le i\le N} e_{i,j} = 1$, find the maximum value of $\sum_{1\le i,j\le N} w_{i,j} e_{i,j}$

Dual formulation: weighted bipartite minimum vertex cover. Constraints: $v_i^l + v_j^r \ge w_{i,j}$, find the minimum value of $\sum_{i=1}^N v_i^l + \sum_{j=1}^N v_j^r$.

Problem 21 (1453 Queen)**.** If $N = 1$, then the answer is $S - 1$. Otherwise as a rook, the queen can move to $N(S-1)$ cells; as a bishop, assume that the queen is moving in direction $\vec{d} = (d_1, ..., d_N)$, $d_i = \pm 1$, and its original position is $\vec{c} = (c_1, ..., c_N)$, we want to calculate the range of $t \ge 1$ such that $\vec{c} + t\vec{d} \in [S]^N$ and we may sort c_i in ascending order. Assume that $I = \{i, d_i = 1\} = \{i_1, ..., i_k, i_1 < i_2 ... < i_k\}$, $J = \{j, d_j = -1\} = \{j_1, ..., j_l, j_1 < j_2 ... < i_l\}$, then the number of possible values of t is $\min_{i\in I}\{s - c_i\} \vee \min_{i \in J}\{c_i - 1\}$ where $\vee$ denotes the minimum operation. So the answer is

$$Answer = Ans_{rook} + Ans_{bishop}, \quad Ans_{rook} = N(S-1),$$

$$Ans_{bishop} = \sum_{I\subset[N]} \min_{i\in I}\{s - c_i\} \vee \min_{i\in J}\{c_i - 1\} = \sum_{I\subset[N],\, |I|\ne 0,N} (S - c_{i_k}) \vee (c_{j_1} - 1) + c_1 - 1 + S - c_N,$$

We only consider the case when $k, l \ge 1$, and calculate the number of cases such that $i_k = u, j_1 = v$ for given $1 \le u \ne v \le N$. 1) $u > v$, then the number of such cases is 2^{u-v-1}; 2)

$v = u + 1, 1 \le u \le N - 1$, the number of such cases is 1. So we may rewrite the above formula as follows:

$$Ans_{bishop} = \sum_{1 \le v < u \le N} 2^{u-v-1}(S - c_u) \vee (c_v - 1) + \sum_{v=u+1, 1 \le u \le N-1} (S - c_u) \vee (c_v - 1) + c_1 - 1 + S - c_N,$$

Direct calculation yields an $O(n^2)$ algorithm, and my submission implementing this algorithm got TLE-13. An improved algorithm divides all the points $(u, v), 1 \le u \le N, 1 \le v \le u - 1$ into two regions. Region 1 on the upper right: $S - c_u < c_v - 1$, and region 2 on the lower left: $S - c_u \le c_v - 1$. We may use the monotonicity of $\{c_i\}$ to calculate these two regions. It finds $vmax$ of region 2 for each fixed u from column $u = N$ downto $u = 1$. Initially we set $vmax = 0$. Column summation happens after determining $vmax$ for a column,

$$ans += (S - c_u) \sum_{v=vmax}^{u-1} 2^{u-v-1} = (S - c_u)(1 + ... + 2^{u-vmax-2}) = (S - c_u)(2^{u-vmax-1} - 1),$$

Row summation happens after each time when $vmax += 1$,

$$ans += (c_{vmax} - 1) \sum_{u'=vmax+1}^{u} 2^{u'-vmax-1} = (c_{vmax} - 1)(2^{u-vmax} - 1),$$

Problem 22 (1455 Freedom of Speech)**.** I used breadth first search at first, but later changed it to Dijkstra's shortest path algorithm. Vertices are (i, j), $0 \le j \le len(s[i])$, indicating that the last term is $s[i]$ with matched length j. An additional vertex *terminal* is added indicating that we've found the two required strings. There are two types of edges: i) $k + len(s[j]) < len(s[i])$, $type = 0$, ii) $k + len(s[j]) \ge len(s[i])$, $type = 1$, Wrote modified KMP algorithm to compute edges of the graph. Direct edges from $(i, 0)$ to terminal are excluded.

Problem 23 (1460 Wires)**.** Claim: 1) auxiliary points has degree 3, and their 3 adjacent edges has pairwise angle $\frac{2\pi}{3}$. This can be proved by calculus of variations.

2) It is impossible to have 3 auxiliary points. Otherwise the total degree of vertices is at least 13, contradiction.

3) The trilinear coordinate of the Fermat point (actually the first isogonic center) of a given triangle ABC is $\sec(A - \frac{\pi}{6}) : \sec(B - \frac{\pi}{6}) : \sec(C - \frac{\pi}{6})$, its barycentric coordinate is $\frac{a}{\cos(A-\frac{\pi}{6})} : \frac{b}{\cos(B-\frac{\pi}{6})} : \frac{c}{\cos(C-\frac{\pi}{6})}$, and I used this formula in computer program calculation.

3) We may enumerate every possible configurations: when there are no auxiliary points, we calculate its minimal spanning tree using Prim's algorithm. When there is only one auxiliary point, this point is uniquely determined by the three points it connects to. When there are two auxiliary points, the configuration is uniquely determined by the permutation of $ABCD$. $auxpt1$ is the Fermat point of $A, B, auxpt2$, $auxpt2$ is the Fermat point of $C, D, auxpt1$.

Problem 24 (1464 Light)**.** Sort vertices according to their polar angles. A container is used to store segments in current region. Insertion or deletion are executed on boundary rays of regions. Comparator is dynamic during the process of sweeping, it is represented by the distance from lamp to the intersections of current region's bisector with segments. Supports deletion by key, query the nearest segment in current region, value is not needed since it's dynamic while sweeping, so I use std::set with delicately designed custom comparator. The comparator compares the distances from the lamp to the intersections of the region bisector and segments.

Problem 25 (1467 Sum of Degrees)**.** Rewrite operator+ in BigNumber struct, and fixed some bug in operator/. Accepted using python, c++ version using BigInteger struct is correct, but its running time is too long. Why?

Used Chu-Vandemonde identity: $\sum_{i=1}^{N}\binom{N}{k} = \binom{N+1}{k+1}$. In practice the rationals' nominators and denominators can be very large.

Problem 26 (1475 Ryaba Hen)**.** Let e_x be the direction parallel to the roof, e_y perpendicular to e_x and points upwards. Then in the direction of e_x, the egg's motion is uniformly accelerated with acceleration $g\sin\theta$, $\tan\theta = \frac{H}{l}$. In the direction of e_y, the egg's motion is periodic with period $T = \frac{2v_0}{g}$. In each peroid $[(n-1)T, nT]$, v_y is uniformly accerlerated with acceleration $-g\cos\theta$ and initial velocity $v_0\cos\theta$. Accepted using python, Qifeng Chen's BigNumber structs and mine. It is worthwhile to note that I wrote a stress unit test for the first time for BigNumber struct. Optimized multiplication in my BigNumber struct and implemented Karatsuba algorithm. Fixed a bug in Qifeng Chen's BigNumber division.

$$v_0\sin\theta t + \frac{1}{2}g\sin\theta t^2 > \sqrt{H^2+l^2}, \quad t = \frac{2nv_0}{g},$$

$$\sin\theta = \frac{H}{\sqrt{H^2+l^2}}, \quad v_0 = \sqrt{2gh}, \quad n(n+1) > \frac{H^2+l^2}{4Hh},$$

Problem 27 (1476 Lunar Code)**.** Dynamic programming from left to right. Consider the first i_0+1 columns, $1 \le i_0 \le N-1$, let $dp[a]$ be the number of 01 matrices of size $M*(i_0+1)$, such that for every $1 \le i \le i_0$,

$$|\{j,\ A[j,i]=0,\ A[j,i+1]=1\}| \le K,$$

The i_0+2-th column has 2^M possibilities, state transition is done by enumerate the number of $00, 01, 10, 11$ substrings in the i_0+1-th and i_0+2-th columns.

Problem 28 (1482 Triangle Game)**.** Notice that the original formula of inverse of $2*2$ matrix has a bug.

$$T_\theta = \begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}, \quad \begin{pmatrix}x_0 \\ y_0\end{pmatrix} = (I - T_\theta)\begin{pmatrix}x_1 \\ y_1\end{pmatrix},$$

New discovery is that operator= is automatically constructed in a custom struct.

Problem 29 (1503 Polynomial)**.** Implemented polynomial division with remainder for the first time. The following partition is obtained by calculate $\gcd(p, p')$ recursively.

$$p = p_1p_2...p_k, \quad p_{i+1}|p_i, \quad p_i \textit{ has no multiple roots},$$

Since each p_i has no multiple roots, we can run Newton's iteration to get all its real roots.

Problem 30 (1511 Fiscal Operations)**.** Given positive integers A, B, C, modify them to A_1, B_1, C_1 such that $A_1 + B_1 = C_1$ with the minimum ℓ_1 cost. Dynamic programming from right to the left. Whether carry-over happens on the i-th digit are considered separatedly. Assume that

$$A = \overline{a_n...a_0}, \quad B = \overline{b_m...b_0}, \quad C = \overline{c_k...c_0}, \quad A_1 = \overline{a'_n...a'_0}, \quad B_1 = \overline{b'_m...b'_0}, \quad C_1 = \overline{c'_k...c'_0},$$

then we have

$$|a_0 - a'_0| + |b_0 - b'_0| + |c_0 - c'_0| \ge |a_0 + b_0 - c_0|, \quad a'_0 + b'_0 = c'_0,$$

$$|a_0 - a'_0| + |b_0 - b'_0| + |c_0 - c'_0| \ge |a_0 + b_0 - c_0 - 10|, \quad a'_0 + b'_0 = c'_0 + 10,$$

Assume that $n \geq m$, and let $dp0[i]$ store the cost from the 0-th digit to the i-th digit with no carry-over happen on the i-th digit, let $dp1[i]$ store the cost from the 0-th digit to the i-th digit with carry-over happen on the i-th digit. State transition equation when $i \leq m$ is

$$dp0[i] = \min(dp0[i-1] + |a_i + b_i - c_i|, dp1[i-1] + |a_i + b_i + 1 - c_i|),$$

$$dp1[i] = \min(dp0[i-1] + |a_i + b_i - c_i - 10|, dp1[i-1] + |a_i + b_i - c_i - 9|),$$

State transition equation when $i > m$ is

$$dp0[i] = \min(dp0[i-1] + |a_i - c_i|, dp1[i-1] + |a_i + 1 - c_i|), \quad dp1[i] = dp1[i-1] + 9 - a_i + c_i,$$

Problem 31 (1531 Zones on a Plane)**.** All the right-angled isosceles triangles can be classified into 8 classes according to the direction of their right angles. Raw calculation is used to help me finding the pattern of the answers, and I found the following pattern. If $n = 1$, then $answer = 1$. If $n = 2m$ is even, then $answer = 2^{m+2} - 4$. If $n = 2m + 1$ is odd, then $answer = 3 \cdot 2^{m+1} - 4$. How to prove it? It remains to be a problem, but I don't want to solve it now.

Problem 32 (1554 Multiplicative Functions)**.** Given a multiplicative function $F : [N] \to \mathbb{Z}/2007\mathbb{Z}, 1 \leq N \leq 10000$, find its Dirichlet inverse $G = F^{-1}$. I used the fact that the Dirichlet inverse of a multiplicative function is again multiplicative, so it suffices to calculate $G(q)$ at prime powers $q = p^{\alpha}$.

Definition 1. A multiplicative function is an arithmetic function $F : \mathbb{N} \to \mathbb{C}$ with property that $F(1) = 1$ and whenever a and b are coprime, then $F(ab) = F(a)F(b)$.

If $f, g : \mathbb{N} \to \mathbb{C}$ are two arithmetic functions, the Dirichlet convolution $f * g$ is a new arithmetic function defined by

$$(f * g)(n) = \sum_{d|n} f(d)g(\frac{n}{d}) = \sum_{ab=n} f(a)g(b),$$

This product occurs naturally in the study of Dirichlet series and describes the multiplication of two Dirichlet series in terms of their coefficients.

$$(\sum_{n\geq 1} \frac{f(n)}{n^s})(\sum_{n\geq 1} \frac{g(n)}{n^s}) = \sum_{n\geq 1} \frac{(f * g)(n)}{n^s},$$

For each f having $f(1) \neq 0$, there exists an arithmetic function f^{-1} with $f * f^{-1} = \epsilon$, called the Dirichlet inverse of f.

Problem 33 (1557 Network Attack)**.** Given a undirected graph with possible self-loops and multiple edges, find the number of ways to divide the graph into at least two connected components by removing two edges. It suffices to find all bridges and 2-cuts of the graph. After deleting all the bridges of the graph, there remained some 2-edge-connected components. There is no 2-cut such that its edges lie in different components, so we may deal with the components separately. In each component, there is no degree 1 vertex, and degree 2 vertices are removable.

Problem 34 (1562 GM-pineapple)**.**

$$Answer = \frac{a^2 b}{8} \int_{1-\frac{2(i+1)}{n}}^{1-\frac{2i}{n}} \pi(1 - z^2)dz, \quad 0 \leq i \leq n-1,$$

Problem 35 (1566 Triangular Postcards). If $\triangle PQR$ can be included inside $\triangle ABC$, then there exist a position such that two vertices of $\triangle PQR$ lie on the sides of $\triangle ABC$. Assume that they are P, Q.

1) P, Q lie on the same side of $\triangle ABC$.

2) P, Q lie on different sides. Assume that $P \in CB, Q \in CA, CP = \lambda, CQ = \mu$, then

$$\lambda^2 + \mu^2 - 2\lambda\mu\cos C = r^2 = \sin^2\frac{C}{2}(\lambda+\mu)^2 + \cos^2\frac{C}{2}(\lambda-\mu)^2,$$

$$R = P + \frac{q}{r}\begin{pmatrix}\cos P & -\sin P\\ \sin P & \cos P\end{pmatrix}(Q-P) = \begin{pmatrix}\lambda + \frac{q}{r}(\mu\cos C\cos P - \lambda\cos P - \mu\sin C\sin P)\\ \frac{q}{r}(\mu\cos C\sin P - \lambda\sin P + \mu\sin C\cos P)\end{pmatrix}$$

Constraints are $0 \le \lambda \le a, 0 \le \mu \le b$, and three constraints depicting $R \in \triangle ABC$:

$$y_R \ge 0, \quad x_R\sin C - y_R\cos C \ge 0, \quad BA \times BR = (x_A - x_B)y_R - y_A x_R + y_A x_B \ge 0,$$

This problem can be solved by checking the sign of $f(\lambda,\mu) = \lambda^2 + \mu^2 - 2\lambda\mu\cos C$ on the boundary of the polygon determined by the above 7 constraints.

Method 2: actually we only need to check case 1 in the discussion above. Avoid using trigonometric functions helps improve numerical accuracy.

Problem 36 (1591 Abstract Thinking). Given input integer N, consider N points on a unit circle, find the number of triple chords tuples such that they intersect pairwisely and at least one intersection is strictly inside the circle. The answer is

$$4\binom{N}{4} + \binom{N}{5} + \binom{N}{6}$$

Problem 37 (1594 Aztec Treasure). Calculate the number of domino tilings on a $m \times n$ rectangle grid. Let $m_1 = \lceil\frac{m}{2}\rceil, n_1 = \lceil\frac{n}{2}\rceil$, the formula is given by

$$Z_{m,n}(1,1) = \prod_{j,k=1}^{m_1,n_1}(4\cos^2\frac{\pi j}{m+1} + 4\cos^2\frac{\pi k}{n+1}),$$

where we define $g(h,v)$ to be the number of tilings with h horizontal and v vertical dominoes.

$$Z_{m,n}(x,y) = \sum_{h,v} g(h,v)x^h y^v, \quad h,v \ge 0, \quad 2(h+v) = mn,$$

Swap m, n if necessary to make sure that m is even.

$$Z_{m,n}(1,1) = \prod_{j,k=1}^{m_1,n_1}(4 + 2\cos\frac{2\pi j}{m+1} + 2\cos\frac{2\pi k}{n+1}),$$

Denote $P_{n_1}(x) = \prod_{k=1}^{n_1}(x + 2\cos\frac{2\pi k}{n+1})$, let $x_j = 4 + 2\cos\frac{2\pi j}{m+1}, 1 \le j \le m_1$, then the result $Z_{m,n}(1,1) = \prod_{j=1}^{m_1} P_{n_1}(x_j)$. We may calculate P_{n_1} by induction. 1) n is even, now

$$P_{n_1}(y + \frac{1}{y}) = \prod_{k=1}^{n_1}(y + \frac{1}{y} + 2\cos\frac{2\pi k}{n+1}) = y^{n_1} - y^{n_1-1}... - y^{1-n_1} + y^{-n_1},$$

$$P_1 = x - 1, \quad P_2 = x^2 - x - 1, \quad P_{n_1} = xP_{n_1-1} - P_{n_1-2},$$

and I let $P_0 = P_{-1} = 1$ in my implementation. 2) n is odd, now

$$P_{n_1}(y+\frac{1}{y}) = \prod_{k=1}^{n_1}(y+\frac{1}{y}+2\cos\frac{2\pi k}{n+1}) = \frac{(y^{n+1}-1)(y-1)}{y+1},$$

$$P_1 = x-2, \quad P_2 = x^2-2x, \quad P_{n_1} = xP_{n_1-1} - P_{n_1-2},$$

and I let $P_0 = 0$ in my implementation.

Key point is to implement a struct representing algebraic integers of the form

$$x_0 + \sum_{1\le j\le m1} x_j 2\cos\frac{2\pi j}{m+1}, \quad m = 2m_1,$$

Its nontrivial arithmetic is essentially inside two methods named reduce() and totalreduce(). Notice that taking modulo is admissible since all the terms above are integral.

Theorem 1 (Domino tilings)**.** 1) The number of ways to cover an $m\times n$ rectangle with $\frac{mn}{2}$ dominoes is given by

$$\prod_{j,k=1}^{\lceil\frac{m}{2}\rceil,\lceil\frac{n}{2}\rceil}(4\cos^2\frac{\pi j}{m+1}+4\cos^2\frac{\pi k}{n+1}),$$

2) The number of tilings of an Aztec diamond of order n is $2^{\frac{n(n+1)}{2}}$.

3) The number of tilings of an augmented Aztec diamond of order n with 3 long rows in the middle rather than 2 is $D(n,n)$, a Delannoy number, defined as follows:

$$D(m,n) = \sum_{k=0}^{\min(m,n)}\binom{m+n-k}{m}\binom{m}{k} = \sum_{k=0}^{\min(m,n)}\binom{m}{k}\binom{n}{k}2^k,$$

$$\sum_{m,n=0}^{\infty} D(m,n)x^m y^n = (1-x-y-xy)^{-1}, \quad \sum_{n=0}^{\infty} D(n,n)x^n = (1-6x+x^2)^{-\frac{1}{2}},$$

Problem 38 (1599 Winding Number)**.** Method 1: calculate $\sum_{i=1}^{n}\angle P_iXP_{i+1}$. Resulted in TLE-12.

Method 2: calculate intersection number of the polygon with ray $y = y_X, x \ge x_X$.

Problem 39 (1600 Airport)**.** Solve quadratic equation to get the time t of the first alarm. Pay attention to machine epsilon of floating numbers.

Problem 40 (1620 Clever House)**.** Markov process, for $0 \le i \le N$, assume that v_i indicates the state that there are i lights on, assume that p_i^t is the probability that there are i lights on at time t, then the transition matrix M satisfies $M_{i,i-1} = \frac{i}{N}$, $M_{i,i+1} = 1-\frac{i}{N}$. Initially we have $p_M^0 = 1$. Let e^t be the expectation of how much light-bulbs will be on at time t, then we have

$$e^t = \sum_{0\le j\le N} jp_j^t, \quad e^{t+1} = \sum_{0\le j\le N} jp_j^{t+1} = \sum_{0\le j\le N} j(\frac{j+1}{N}p_{j+1}^t + (1-\frac{j-1}{N})p_{j-1}^t),$$

$$e^{t+1} = \sum_{0\le j\le N}(\frac{j(j-1)}{N}+(j+1)(1-\frac{j}{N}))p_j^t = \sum_{0\le j\le N}(j+1-\frac{2j}{N})p_j^t = (1-\frac{2}{N})e^t+1,$$

$$e^K = (\frac{N-2}{N})^K(M-\frac{N}{2})+\frac{N}{2},$$

Problem 41 (1621 Definite Integral)**.** Roots finding algorithms. Given an integer coefficient degree 4 polynomial with $|a_i| \le 10^6, a_4 \ne 0$. Notice that the precision requirement is high, relative error or absolute error is no more than 10^{-9}.

$$\int_{|x|=\epsilon} \frac{1}{x} dx = \int_0^{2\pi} \frac{e^{-i\theta}}{\epsilon} d\epsilon e^{i\theta} = \int_0^{2\pi} i d\theta = 2\pi i,$$

So residue theorem says that if a meromorphic function $f = \frac{g}{x-x_0}$ where g is holomorphic near x_0, then $\int_{|x-x_0|=\epsilon} f dx = 2\pi i g(x_0)$. Given $P(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$, we want to eliminate a_3 by translation $x' = x + \frac{a_3}{4}$.

$$\begin{aligned}
P'(x') =&P(x) = (x' - \frac{a_3}{4})^4 + a_3(x' - \frac{a_3}{4})^3 + a_2(x' - \frac{a_3}{4})^2 + a_1(x' - \frac{a_3}{4}) + a_0 \\
=&x'^4 + 6x'^2(\frac{a_3}{4})^2 - 4x'(\frac{a_3}{4})^3 + (\frac{a_3}{4})^4 - 3x'^2 a_3 \frac{a_3}{4} + 3x' a_3(\frac{a_3}{4})^2 - a_3(\frac{a_3}{4})^3 \\
&+ a_2x'^2 - 2x'a_2\frac{a_3}{4} + a_2(\frac{a_3}{4})^2 + a_1x' - a_1\frac{a_3}{4} + a_0 \\
=&x'^4 + x'^2(6(\frac{a_3}{4})^2 - 3a_3\frac{a_3}{4} + a_2) + x'(-4(\frac{a_3}{4})^3 + 3a_3(\frac{a_3}{4})^2 - 2a_2\frac{a_3}{4} + a_1) \\
&+ (\frac{a_3}{4})^4 - a_3(\frac{a_3}{4})^3 + a_2(\frac{a_3}{4})^2 - a_1\frac{a_3}{4} + a_0 \\
=&x'^4 + x'^2(-\frac{3a_3^2}{8} + a_2) + x'(\frac{a_3^3}{8} - \frac{a_2a_3}{2} + a_1) - \frac{3a_3^4}{256} + \frac{a_2a_3^2}{16} - \frac{a_1a_3}{4} + a_0,
\end{aligned}$$

So we may define

$$a_2' = -\frac{3a_3^2}{8} + a_2, \quad a_1' = \frac{a_3^3}{8} - \frac{a_2a_3}{2} + a_1, \quad a_0' = -\frac{3a_3^4}{256} + \frac{a_2a_3^2}{16} - \frac{a_1a_3}{4} + a_0,$$

Now we substitute x', P', a' by x, P, a, it becomes

$$P(x) = x^4 + a_2x^2 + a_1x + a_0 = (x^2 - 2ax + b)(x^2 + 2ax + c), \quad a > 0, \quad a^2 < b, c,$$

Assume that it has two roots in the upper half plane $x_1 = -a + ui, x_2 = a + vi, u, v > 0$, then the integral is

$$\begin{aligned}
&\int_{\mathbb{R}} \frac{1}{P(x)} dx = 2\pi i(\frac{1}{(x_1 - \overline{x_1})(x_1 - x_2)(x_1 - \overline{x_2})} + \frac{1}{(x_2 - x_1)(x_2 - \overline{x_1})(x_2 - \overline{x_2})}) \\
=&\frac{\pi}{u(4a^2 - u^2 + v^2 - 4aui)} + \frac{\pi}{v(4a^2 - v^2 + u^2 + 4avi)} = \frac{\pi((4a^2 - u^2 + v^2)/u + (4a^2 - v^2 + u^2)/v)}{16a^4 + 8a^2(u^2 + v^2) + (v^2 - u^2)^2},
\end{aligned}$$

where we used

$$(x_1 - x_2)(x_1 - \overline{x_2}) = (-2a + (u - v)i)(-2a + (u + v)i) = 4a^2 - (u^2 - v^2) - 4aui,$$

$$(x_2 - x_1)(x_2 - \overline{x_1}) = (2a + (v - u)i)(2a + (v + u)i) = 4a^2 - (v^2 - u^2) + 4avi,$$

and the imaginary part is

$$\frac{4au/u}{(4a^2 - u^2 + v^2)^2 + (4au)^2} + \frac{-4av/v}{(4a^2 - v^2 + u^2)^2 + (4av)^2} = 0,$$

$$(4a^2 - u^2 + v^2)^2 + (4au)^2 = (4a^2 - v^2 + u^2)^2 + (4av)^2 = 16a^4 + 8a^2(u^2 + v^2) + (v^2 - u^2)^2,$$

1) When $a_1 = 0$, either one of the following two cases occur: i)$a = 0$, the integral becomes

$$\int_{\mathbb{R}} \frac{1}{(x^2+u^2)(x^2+v^2)}dx = \frac{1}{v^2-u^2}\int_{\mathbb{R}}(\frac{1}{x^2+u^2} - \frac{1}{x^2+v^2})dx = \frac{1}{v^2-u^2}(\frac{\pi}{u} - \frac{\pi}{v}),$$

where we used that

$$\int_{-\infty}^{+\infty} \frac{1}{x^2+a^2}dx = \int_{-\infty}^{+\infty} \frac{1/a}{(\frac{x}{a})^2+1}d\frac{x}{a} = \frac{1}{a}\arctan\frac{x}{a}\Big|_{-\infty}^{+\infty} = \frac{\pi}{a},$$

This result agrees with our previous calculation since

$$\frac{\pi((-u^2+v^2)/u + (-v^2+u^2)/v)}{(v^2-u^2)^2} = \frac{1}{v^2-u^2}(\frac{\pi}{u} - \frac{\pi}{v}),$$

$$P(x) = (x^2+u^2)(x^2+v^2) = x^4 + (u^2+v^2)x^2 + u^2v^2,$$

and thus we can solve a quadratic equation to get values of u, v.

ii)$a > 0, u = v$, the integral becomes

$$\frac{8a^2\pi(1/u)}{16a^4+16a^2u^2} = \frac{\pi}{2u(a^2+u^2)},$$

$$P(x) = (x^2+2ax+a^2+u^2)(x^2-2ax+a^2+u^2) = x^4 + 2(u^2-a^2)x^2 + (a^2+u^2)^2,$$

and thus we can solve a linear equation to get values of a, u.

2) When $a_1 \neq 0$, notice that a^2 is a algebraic number with degree 3.

$$a = \frac{x_2+\overline{x_2}-x_1-\overline{x_1}}{4}, \quad \tilde{a}_1 = \frac{x_1+x_2-\overline{x_1}-\overline{x_2}}{4} = \frac{(u+v)i}{2}, \quad \tilde{a}_2 = \frac{x_1+\overline{x_2}-\overline{x_1}-x_2}{4} = \frac{(u-v)i}{2},$$

$$\tilde{a}_1^2 = -\frac{(u+v)^2}{4}\tilde{a}_2^2 = -\frac{(u-v)^2}{4}$$

Coefficients of P satisfy

$$P(x) = (x^2+2ax+a^2+u^2)(x^2-2ax+a^2+v^2) = x^4+(u^2+v^2-2a^2)x^2+2a(v^2-u^2)x+(a^2+u^2)(a^2+v^2),$$

$$a_2 = u^2+v^2-2a^2, \quad a_1 = 2a(v^2-u^2), \quad a_0 = (a^2+u^2)(a^2+v^2),$$

$$a^2 + \tilde{a}_1^2 + \tilde{a}_2^2 = a^2 - \frac{u^2+v^2}{2} = -\frac{a_2}{2},$$

$$a^2(\tilde{a}_1^2+\tilde{a}_2^2)+\tilde{a}_1^2\tilde{a}_2^2 = -\frac{a^2(u^2+v^2)}{2} + \frac{(u^2-v^2)^2}{16} = -\frac{a^2(u^2+v^2)}{2} + \frac{(u^2+v^2)^2}{16} - \frac{u^2v^2}{4} = \frac{1}{4}((\frac{a_2}{2})^2 - a_0),$$

$$a^2\tilde{a}_1^2\tilde{a}_2^2 = \frac{a^2(u^2-v^2)^2}{16} = \frac{1}{16}(\frac{a_1}{2})^2,$$

$$R(x) = x^3 + b_2x^2 + b_1x + b_0, \quad x' = x + \frac{b_2}{3},$$

$$b_2 = \frac{a_2}{2}, \quad b_1 = \frac{\frac{a_2^2}{4} - a_0}{4}, \quad b_0 = -\frac{a_1^2}{64},$$

$$R(x) = x'^3 + x'(-\frac{b_2^2}{3} + b_1) + \frac{2b_2^3}{27} - \frac{b_1b_2}{3} + b_0 = S(x'),$$

$$S(x) = x^3 + c_1 x + c_0, \quad c_1 = -\frac{b_2^2}{3} + b_1, \quad c_0 = \frac{2b_2^3}{27} - \frac{b_1 b_2}{3} + b_0, \quad x = y - \frac{c_1}{3y}$$

$$T(y) = S(x) = y^3 + c_0 - \frac{c_1^3}{27y^3}, \quad z = y^3, \quad \omega = e^{\frac{2\pi i}{3}},$$

$$x_1 = y_1 + y_2, \quad x_2 = y_1\omega + y_2\omega^2, \quad x_3 = y_1\omega^2 + y_2\omega,$$

Use long double and one step Newton method to improve result's accuracy.

Method 2: Find a square matrix such that $P(x)$ is its characteristic polynomial. According to rational canonical form, we may construct

$$A = \begin{pmatrix} 0 & & & -a_0 \\ 1 & 0 & & -a_1 \\ & 1 & 0 & -a_2 \\ & & 1 & -a_3 \end{pmatrix}, \quad \det(\lambda I - A) = P(\lambda),$$

Then we may apply iterative algorithms that finds unsymmetric eigenvalues of matrix A. But I haven't implemented this idea successfully.

Problem 42 (1625 Hankel Matrix)**.** A Hankel matrix is a matrix of the form

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_n \\ \alpha_2 & \alpha_3 & \alpha_4 & \ldots & \alpha_{n+1} \\ \alpha_3 & \alpha_4 & \alpha_5 & \ldots & \alpha_{n+2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \alpha_n & \alpha_{n+1} & \alpha_{n+2} & \ldots & \alpha_{2n-1} \end{pmatrix}$$

Given the size of the matrix n, find an integer Hankel matrix of size n with all non-negative elements and with determinant equal to one. Moreover, all its square submatrices containing upper left cell must also have determinant equal to one. Refering to *acm.timus.ru/forum/thread.aspx?id=21158*, my output is the sequence of Catalan numbers. Let

$$\alpha_1 = 1, \quad \alpha_2 = 1, \quad \alpha_3 = 2, \quad \alpha_4 = 5, \quad \alpha_5 = 14, ..., \alpha_n = \sum_{i=1}^{n-1} \alpha_i \alpha_{n-i}, \quad n \geq 2,$$

Define the generating function $f(x) = \sum_{n=1}^{\infty} \alpha_n x^n$, then by the recurrence relation above, we have the functional equation

$$f(x) = f(x)^2 + x, \quad f(x) = \frac{1 - \sqrt{1-4x}}{2},$$

The sign in front of $\sqrt{1-4x}$ is chosen in the way that $f(x)$ is monotonic increasing w.r.t. x.

$$\sqrt{1-4x} = \sum_{n \geq 0} \binom{\frac{1}{2}}{n} (-4x)^n, \quad \binom{\frac{1}{2}}{n} = (-1)^{n-1} \frac{(2n-3)!!}{n! 2^n},$$

$$\sqrt{1-4x} = -\sum_{n \geq 0} \frac{(2n-3)!! 2^n}{n!} x^n = 1 - 2\sum_{n \geq 1} \frac{(2n-2)!}{n!(n-1)!} x^n,$$

So we know that

$$f(x) = \sum_{n \geq 1} \frac{(2n-2)!}{n!(n-1)!} x^n = \sum_{n \geq 1} \frac{1}{n} \binom{2n-2}{n-1} x^n, \quad \alpha_n = \frac{1}{n} \binom{2n-2}{n-1},$$

Theorem 2 (Catalan Hankel determinants)**.** For the Catalan numbers $\{\alpha_n\}_{n\geq 1} = \{1, 1, 2, 5, 14, ...\}$, $\alpha_{n+1} = \frac{1}{n+1}\binom{2n}{n}$, let $C_n^t = (\alpha_{i+j+t+1})_{0\leq i,j\leq n-1}$ denote the Hankel matrix. Then for $n \geq 1$, the following identities hold:

$$\det C_n^0 = \det C_n^1 = 1, \quad \det C_n^2 = n+1, \quad \det C_n^3 = \frac{(n+1)(n+2)(2n+3)}{6},$$

Proof. 1) We focus on $t = 0$ case firstly, and give the Cholesky decomposition $A = C^t C$ of matrix $A = C_n^0$, where C is upper triangular and has positive diagonals. Let $C = (c_{ij})_{0\leq i,j\leq n-1}$, then its elements are given by $c_{ij} = \frac{2i+1}{i+j+1}\binom{2j}{j-i}$. Notice that $c_{ij} = 0$ when $i > j$. It suffices to prove the following combinatorial identity, which is equivalent to $\sum_k c_{ki}c_{kj} = \alpha_{i+j+1}$:

$$\sum_k \frac{(2k+1)^2}{(k+i+1)(k+j+1)}\binom{2i}{i-k}\binom{2j}{j-k} = \frac{1}{i+j+1}\binom{2i+2j}{i+j},$$

$$\frac{2k+1}{k+i+1}\binom{2i}{i-k} = (\frac{1}{i-k} - \frac{1}{k+i+1})\frac{(2i)!}{(i-k-1)!(i+k)!} = \binom{2i}{i-k} - \binom{2i}{i-k-1}$$
$$= coeff < x^{i-k}, (1+x)^{2i}(1-x) >,$$
$$\frac{2k+1}{k+j+1}\binom{2j}{j-k} = \binom{2j}{j-k} - \binom{2j}{j-k-1} = coeff < x^{k-j}, (1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >,$$

Without loss of generality, we assume that $0 \leq i \leq j$, then the range of k is $0 \leq k \leq i$. We perform the following decomposition:

$$coeff < x^{i-j}, (1+x)^{2i}(1-x)(1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >= I + II,$$

$$I \triangleq \sum_{0\leq l\leq i} coeff < x^l, (1+x)^{2i}(1-x) > coeff < x^{i-j-l}, (1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >,$$

$$II \triangleq \sum_{i+1\leq l\leq 2i+1} coeff < x^l, (1+x)^{2i}(1-x) > coeff < x^{i-j-l}, (1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >,$$

We get the following identity from the calculation above.

$$\sum_k \frac{(2k+1)^2}{(k+i+1)(k+j+1)}\binom{2i}{i-k}\binom{2j}{j-k} = I,$$

On the other hand, we can show that $I = II$ as follows. Let $p = 2i+1-l$, we have

$$coeff < x^l, (1+x)^{2i}(1-x) >= coeff < x^{-l}, (1+\frac{1}{x})^{2i}(1-\frac{1}{x}) >= coeff < x^p, (x+1)^{2i}(x-1) >,$$

$$coeff < x^{i-j-l}, (1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >= coeff < x^{j-i+l}, (1+x)^{2j}(1-x) >$$
$$= coeff < x^{i-j-p}, (\frac{1}{x}+1)^{2j}(\frac{1}{x}-1) >,$$

In the last step we used $j - i - l - (2j+1) = i - j - p$. So we see that $I = II$ is indeed true. Finally we have

$$I + II = coeff < x^{i-j}, (1+x)^{2i}(1-x)(1+\frac{1}{x})^{2j}(1-\frac{1}{x}) >$$
$$= coeff < x^{i+j}, (1+x)^{2i+2j}(2-x-\frac{1}{x}) >= 2\binom{2i+2j}{i+j} - \binom{2i+2j}{i+j-1} - \binom{2i+2j}{i+j+1}$$
$$= \binom{2i+2j}{i+j}(2 - \frac{2(i+j)}{i+j+1}) = \frac{2}{i+j+1}\binom{2i+2j}{i+j},$$

So the desired identity holds since $LHS = I = \frac{1}{2}(I + II)$.

2) Secondly, when $t = 1$, let $A = C_n^1$. Assume its Cholesky decomposition is $A = C^tC$, $C = (c_{ij})_{0\leq i,j\leq n-1}$, then its elements are given by $c_{ij} = \frac{i+1}{j+1}\binom{2j+2}{j-i}$. It suffices to prove that $\sum_k c_{ki}c_{kj} = \alpha_{i+j+2}$, which is equivalent to the following combinatorial identity:

$$\sum_k \frac{(k+1)^2}{(i+1)(j+1)}\binom{2i+2}{i-k}\binom{2j+2}{j-k} = \frac{1}{i+j+2}\binom{2i+2j+2}{i+j+1},$$

$$\frac{k+1}{i+1}\binom{2i+2}{i-k} = (\frac{1}{i-k} - \frac{1}{i+1})\frac{(2i+2)!}{(i+k+2)!(i-k-1)!} = \binom{2i+2}{i-k} - 2\binom{2i+1}{i-k-1}$$
$$= coeff < x^{i-k}, (1+x)^{2i+2} - 2x(1+x)^{2i+1} >= coeff < x^{i-k}, (1-x)(1+x)^{2i+1} >,$$

$$\frac{k+1}{j+1}\binom{2j+2}{j-k} = coeff < x^{k-j}, (1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >,$$

We assume that $0 \leq i \leq j$, then the range of k is $0 \leq k \leq i$.

$$coeff < x^{i-j}, (1-x)(1+x)^{2i+1}(1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >= I + II,$$

$$I \triangleq \sum_{0\leq l\leq i} coeff < x^l, (1-x)(1+x)^{2i+1} > coeff < x^{i-j-l}, (1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >,$$

$$II \triangleq \sum_{i+2\leq l\leq 2i+2} coeff < x^l, (1-x)(1+x)^{2i+1} > coeff < x^{i-j-l}, (1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >,$$

It is easy to see that $coeff < x^{i+1}, (1-x)(1+x)^{2i+1} >= 0$, and we have the following identity:

$$\sum_k \frac{(k+1)^2}{(i+1)(j+1)}\binom{2i+2}{i-k}\binom{2j+2}{j-k} = I,$$

Similar to the case $t = 0$, we can show that $I = II$ as follows. Let $p = 2i + 2 - l$, we have

$$coeff < x^l, (1-x)(1+x)^{2i+1} >= coeff < x^p, (x-1)(x+1)^{2i+1} >,$$

$$coeff < x^{i-j-l}, (1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >= coeff < x^{i-j-p}, (\frac{1}{x}-1)(\frac{1}{x}+1)^{2j+1} >,$$

So we see that $I = II$ is true again. Finally we have

$$I + II = coeff < x^{i-j}, (1-x)(1+x)^{2i+1}(1-\frac{1}{x})(1+\frac{1}{x})^{2j+1} >$$
$$= coeff < x^{i+j+1}, (1+x)^{2i+2j+2}(2 - x - \frac{1}{x}) >= 2\binom{2i+2j+2}{i+j+1} - \binom{2i+2j+2}{i+j} - \binom{2i+2j+2}{i+j+2}$$
$$= \binom{2i+2j+2}{i+j+1}(2 - \frac{2(i+j+1)}{i+j+2}) = \frac{2}{i+j+2}\binom{2i+2j+2}{i+j+1},$$

And $LHS = I = \frac{1}{2}(I + II)$ follows. □

Problem 43 (1626 Interfering Segment)**.** Reference: Computational geometry - algorithms and applications, chapter 2, line segment intersection. Introduction to algorithms 3rd edition, chapter 33, determining whether any pair of segments intersects.

Assume that the polygon is $P_1P_2...P_n$. $\triangle ABC$ is part of a legal triangulation $\iff$ each of AB, BC, AC divides the polygon into two parts. Only if part is trivial. If part: assume that $\Omega_A, \Omega_B, \Omega_C$ are the parts of the polygon that don't contain A, B, C after divided by BC, CA, AB respectively, then each of them is a polygon without self intersection. So $\Omega_A, \Omega_B, \Omega_C$ can be triangulated, adding $\triangle ABC$ to obtain a triangulation of the original polygon.

$$X \in \triangle ABC \iff X \notin \Omega_A, \Omega_B, \Omega_C,$$

Preprocessing: for any pair i, j not next to each other, judge if P_iP_j intersect with any edge besides $P_{i-1}P_i, P_iP_{i+1}, P_{j-1}P_j, P_jP_{j+1}$, store these boolean variables in $flag[i][j]$. If $flag[i][j] == true$, judge if X is on the $P_{i+1}...P_{j-1}$ side or $P_{j+1}...P_{i-1}$ side of P_iP_j, or is exactly on P_iP_j, and store it in $sideX[i][j]$. $flag[i][i+1]$ is always true. Denote by l_X the ray $y = y_X, x \geq x_X$, fix i and let j iterate from $i+1$ to $i-1$, we may maintain the intersection number of l_X with $P_iP_{i+1}...P_j$, and thus know $sideX[i][j]$. Similarly we get $sideY[i][j]$.

Problem 44 (1637 Triangle Game 2)**.** 1) Oriented transform: 0 or 2;

2) Non-oriented transform: 1 or 3. Reflector about the axis $\mathbb{R}(\cos\theta, \sin\theta)$ is given by

$$T_\theta = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto T_\theta \begin{pmatrix} x \\ y \end{pmatrix}$$

Problem 45 (1659 Regular Triangles)**.** Pattern is first found in a unit circle.

Problem 46 (1660 The Island of Bad Luck)**.** Method 1: calculating Apollonian circles, but I didn't use this method. 1) Assume that

$$l_0 : y = y_0, \quad l_1 : y = y_1, \quad y_0 > y_1, \quad l_2 : x^2 + (y - y_1 - r_2)^2 = r_2^2, \quad r_2 < \frac{y_0 - y_1}{2},$$

Find the equation of circle ω such that ω is tangent to l_0, l_1, l_2. $r_\omega = \frac{y_0 - y_1}{2}$. There are two solutions,

$$\omega_1 : (x - 2\sqrt{r_2 r_\omega})^2 + (y - \frac{y_0 + y_1}{2})^2 = r_\omega^2,$$

$$\omega_2 : (x + 2\sqrt{r_2 r_\omega})^2 + (y - \frac{y_0 + y_1}{2})^2 = r_\omega^2,$$

where ω_1 is on the left, ω_2 is on the right.

2) In the original problem, assume that the large circle and small circle are Γ_0, Γ_2, victim's circle is Γ_1. We've already known that Γ_0 has radius R, Γ_2 has radius r, distance of their centers is d. Assume that Γ_0, Γ_1 are tangential at $P(0, R)$, and let it be the inversion center with radius $\sqrt{2}R$. More precisely, the inversion is

$$\varphi : (x, y) \mapsto (\frac{2R^2 x}{x^2 + (y - R)^2}, \frac{2R^2(y - R)}{x^2 + (y - R)^2} + R),$$

$$\Gamma_0 \mapsto l_0 : y = 0, \quad \Gamma_1 \mapsto l_1 : y = R - \frac{R}{r_1^2},$$

Assume that the center of the small circle is $(d\sin\theta, d\cos\theta)$, then the equation of Γ_2 is

$$\Gamma_2 : (x - d\sin\theta)^2 + (y - d\cos\theta)^2 = r^2,$$

$$l_2 = \varphi(\Gamma_2) : (\frac{2R^2 x}{x^2 + (y - R)^2} - d\sin\theta)^2 + (\frac{2R^2(y - R)}{x^2 + (y - R)^2} + R - d\cos\theta)^2 = r^2,$$

$$\frac{4R^4}{x^2+(y-R)^2} + \frac{2R^2(y-R)(R-d\cos\theta) - 2R^2xd\sin\theta}{x^2+(y-R)^2} + R^2 - 2Rd\cos\theta + d^2 - r^2 = 0,$$

I didn't implement this method since Möbius transformation formulas turned out to be too complicated without using complex analysis.

Method 2: we only consider the case when $\theta = 0$.

$$\Gamma_2 : x^2 + (y-d)^2 = r^2, \quad \varphi(\Gamma_2) = \Gamma_4,$$

All the possible circle chain configurations can be regarded as the orbit of a particular configuration under the action of $SO(2)$. When $d = 0$, the action is exactly rotation around the common center of the two circles. When $d \neq 0$, assume that the center of the small circle is $(0, d)$, then there is a transformation that fixes Γ_0 and takes Γ_2 to a circle centered at the origin. More precisely, find the circle Γ_3 which is orthogonal to both Γ_1, Γ_2. Its center is on the radical axis l_{rad} of Γ_0, Γ_2, with radius equals to length of tangents, so its equation is

$$l_{rad} : 2dy = R^2 + d^2 - r^2, \quad \Gamma_3 : x^2 + (y - \frac{R^2+d^2-r^2}{2d})^2 = (\frac{R^2+d^2-r^2}{2d})^2 - R^2,$$

Using complex analysis, an orientation preserving auto morphism of the unit disk is

$$f : z \mapsto \frac{z-w}{1-\overline{w}z}, \quad |w| < 1, \quad f'(z) = \frac{1-w\overline{w}}{(1-\overline{w}z)^2}, \quad f^{-1}(z) = \frac{z+w}{1+\overline{w}z},$$

Let $w = x_0 + y_0 i, z = x + yi$, the formula of this transformation is

$$\begin{aligned} f(x+yi) &= \frac{(x-x_0)+(y-y_0)i}{1-x_0x-y_0y+(y_0x-x_0y)i} = \frac{(z-w)(1-w\overline{z})}{(1-x_0x-y_0y)^2+(y_0x-x_0y)^2} \\ &= \frac{x+yi+(x_0^2-y_0^2+2x_0y_0i)(x-yi)-(x_0+y_0i)(1+x^2+y^2)}{1-2(x_0x+y_0y)+(x_0^2+y_0^2)(x^2+y^2)} \end{aligned}$$

It becomes too long, so I used its complex form in my implementation. In this scenario, we change the disk radius to R and let $\Re(w) = 0$. We have

$$f(z) = \frac{z-w}{1-\frac{\overline{w}z}{R^2}}, \quad w = (\frac{R^2+d^2-r^2}{2d} - \sqrt{(\frac{R^2+d^2-r^2}{2d})^2 - R^2})i$$

Since $f(w) = 0, f'(w) \in \mathbb{R}$ we know that f maps Γ_3 to $\mathbb{R}$. We may regard it as a translation in the group of hyperbolic automorphisms of Γ_0. Denote the first and the last circles by Ω_A, Ω_B. Observe that the required minimum distance between Ω_A, Ω_B is obtained when they are symmetric along the y-axis. Γ_4 is the image of Γ_2 by translation, assume that it has radius r_4. The image of the first circle by translation is

$$|z - z_A| = \frac{R - r_4}{2}, \quad z_A = r_4 \sin\frac{\delta\theta}{2} + i\cos\frac{\delta\theta}{2},$$

where $\delta\theta$ is the angle between centers of $f(\Omega_A), f(\Omega_B)$. It is an invariant since Γ_0, Γ_4 are concentric. In my program, three points are selected on $f(\Omega_A)$, and Ω_A is obtained by finding the circumcircle of preimages of the three points.

Problem 47 (1661 Dodecahedron)**.** The symmetry group of dodecahedron in $SO(3)$ is A_5. Assume that its edges are $e_1, ..., e_{30}$. Given $c_1, ..., c_{30} \in [30] = \{1, 2, ..., 30\}$, find the number of different dodecahedra. A coloring is to give each edge e_i a color $c[s_i]$, where $s \in S_{30}$ is a permutation.

Two coloring s^1, s^2 are identical means there exists $\sigma \in A_5 \subset S_{30}$, such that for any $1 \leq i \leq 30$, $c[s^2(\sigma(i))] = c[s^1(\sigma(i))]$. Ignoring c and A_5, all the possible color assignments can be regarded as the permutation group S_{30}. There is a subgroup G of S_{30} determined by c, such that

$$c \circ s^2 = c \circ s^1 : [30] \to [30] \iff \text{exists } g \in G, g \circ s^2 = s^1, \quad s^1, s^2 \in S_{30}, \quad G = \prod_{x \in [30]} S_{c^{-1}(x)},$$

G is the product of permutation groups on each fiber of c. So S_{30} is given a double coset structure $G \curvearrowright S_{30} \curvearrowleft A_5$, and we are asked to calculate its cardinality.

The orbits of $S_{30} \curvearrowleft A_5$ are S_{30}/A_5, the set of right cosets of A_5 in S_{30}. Similarly, the orbits of $G \curvearrowright S_{30}$ are $G\backslash S_{30}$, the set of left cosets of G in S_{30}. We may assume that the image of c is $1, 2, ..., k$, and $|c^{-1}(i)| = n_i, \sum_{1\leq i\leq k} n_i = 30$. Our task is to calculate $|G\backslash S_{30}/A_5|$.

$X = G\backslash S_{30}$ can be described as all the sequences $d_1, ..., d_{30}$ in which i appear n_i times, $1 \leq i \leq k$. Burnside's lemma says that

$$|X/A_5| = \frac{1}{|A_5|} \sum_{\sigma \in A_5} |X^\sigma|, \quad X^\sigma = \{x \in X, \sigma(x) = x\},$$

Elements of A_5 can be divided into 4 classes: identity, 15 order 2 elements, 20 order 3 elements, 24 order 5 elements. Let $\sigma_l \in A_5$ be an element with order l. i) $l = 1, 3, 5$, then $|X^{\sigma_l}| = \frac{(\frac{30}{l})!}{\prod(\frac{n_i}{l})!}$ when l divides each n_i, otherwise $|X^{\sigma_l}| = 0$. ii) $l = 2$, if the number of odds in n_i is larger than 2, $|X^{\sigma_l}| = 0$; if there are odds in n_i then $|X^{\sigma_l}| = \frac{(\frac{30}{l})!}{\prod(\frac{n_i}{l})!}$; otherwise there are 2 odds in n_i, $|X^{\sigma_l}| = \frac{2*(14)!}{\prod(\lfloor\frac{n_i}{l}\rfloor)!}$. Used Qifeng Chen's BigInteger struct for large number calculation, and made some modifications. One precious experience is that, never ever inherit from any std:: type, except for the standard types you're supposed to inherit from. Accepted using both implementations of BigInteger of mine and Qifeng Chen's.

Proof. Assume $G \curvearrowright X$ is a left group action.

$$X^g = \{x \in X, g(x) = x\}, \quad G^x = \{g \in G, g(x) = x\}, \quad |\text{orbit}(x)| = \frac{|G|}{|G^x|},$$

$$\frac{1}{|G|} \sum_{g \in G} |X^g| = \frac{1}{|G|} \sum_{x \in X} |G^x| = \sum_{x \in X} \frac{1}{|\text{orbit}(x)|} = |X/G|,$$

□

Problem 48 (1662 Goat in the Garden 6)**.** Note that the bed is a convex polygon $P_1P_2...P_n$. Let $D_r = \bigcup_{p\in polygon} \odot(p, r)$, we need to determine whether $\bigcap_{p\in polygon} \odot(p, R) \cap \overline{D_r}$ is nonempty. $\bigcap_{p\in polygon} \odot(p, R) = \bigcap_{1\leq i\leq n} \odot(P_i, R)$ is an convex set. For any $p \notin D_r$, let q be its projection onto ∂D_r. If $p \in \bigcap_{1\leq i\leq n} \odot(P_i, R)$, then $q \in \bigcap_{1\leq i\leq n} \odot(P_i, R)$, so it sufficies to check if $\partial D_r \cap \bigcap_{1\leq i\leq n} \odot(P_i, R)$ is nonempty. ∂D_r can be divided into n arcs and n segments. 1) Segment circles intersection.

2) For an arc $\overline{AB} \in \odot(O, r)$ and $\odot(X, R)$, i) Both $AX > R, BX > R$ hold, then $\overline{AB} \cap \odot(X, R)$ is empty since O, X lie on the same side of AB. ii) $OX \leq R - r$, check the next circle. iii) $OX > R - r$, get intersection C, D and judge that if $\overline{AB}$ split into two parts. Assume μ is the signed length of OH the direction of XH, $\nu = HC$, then

$$R^2 - r^2 = XH^2 - OH^2 = 2\mu XO + XO^2, \quad \nu = \sqrt{r^2 - \mu^2},$$

Problem 49 (1668 Death Star 2)**.** $A_{N\times M}, b_N$, find x_M such that $\|Ax-b\|_2^2$ reaches minimum. If the solution is ambiguous, output the one that $\|x\|_2^2$ is the minimum. Let $A = UsV^t$ be the singular value decomposition of A,

$$s^{inv}[i] = \begin{cases} \frac{1}{s[i]}, s[i] > 0, \\ 0, s[i] = 0. \end{cases}, \quad A^{inv} = V s^{inv} U^t$$

Then we claim that $x = A^{inv}b$. While calculating SVD, we use Golub-Kahan bidiagonalization in phase 1. Householder reflection is given by

$$x = A_{k:m,k}, \quad v_k = \mathrm{sign}(x_1)\|x\|_2 e_1 + x, \quad v_k = \frac{v_k}{\|v_k\|_2}, \quad A_{k:m,k:n} -= 2v_k(v_k^* A_{k:m,k:n})$$

Householder reflector: $x \mapsto Fx = \pm\|x\|e_1$. Givens rotation acting on the i,j-th rows or columns: $G(i,j,\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$. How to use Givens rotations to eliminate the off-diagonal elements? Actually the remaining steps after bidiagonalization only use Givens rotations.

Problem 50 (1675 Lunar Code 2)**.**

$$2^{mn} = \sum_{k,l=0}^{m-1,n-1} \binom{m}{k}\binom{n}{l} F(m-k, n-l) + 1, \quad F(0,0) = 1,$$

$$f(x,y) = \sum_{k,l} F(k,l)x^k y^l, \quad 2^{mn} = coeff < 1, f(1+x^{-1})^m(1+y^{-1})^n >,$$

A more efficient way is to use the following identity which calculates the number of matrices with each rows not all zero:

$$\sum_{l=1}^{n} \binom{n}{l} F(m,l) = (2^n - 1)^m,$$

Assume that $y_n = F(m,n), b_n = (2^n-1)^m$, we want to solve the inverse problem of the linear system above. The system and its solution are given by

$$\sum_{l=1}^{n} \binom{n}{l} y_l = b_n, \quad y_n = \sum_{l=0}^{n-1} (-1)^l \binom{n}{l} b_{n-l},$$

Its proof is essentially the following combinatorial identity:

$$\sum_{1 \le i+l=m \le n-1} \binom{n}{i}\binom{n-i}{l}(-1)^l = coeff < x^m, (1+x)^n(1+x)^{-n} >= 0,$$

Python program is correct but got TLE-7, so I wrote a c++ program in addition. Another attempt is to use generating function:

$$f(x) = \sum_{n\ge1} y_n x^n, \quad g(u) = \sum_{n\ge1} b_n u^n, \quad b_n = coeff_x < x^n, f(x)(1+x)^n >,$$

$$g(u) = coeff_x < 1, \sum_{n\ge1}(1+x^{-1})^n f(x)u^n >,$$

$$y_n = coeff_{u'} < u'^n, (1-u')^n g(u') >, \quad f(x) = coeff_{u'} < 1, \sum_{n\geq 1} (u'^{-1}-1)^n g(u') x^n >,$$

$$g(u) = coeff_{u',x} < 1, g(u') \sum_{m\geq 1} (u'^{-1}-1)^m x^m \sum_{n\geq 1} (1+x^{-1})^n u^n >,$$

We may consider such a scenario that $x, u, u' \in \mathbb{C}$, $1 > |u'| > |x| > |u| > 0$, such that the following two series are convergent:

$$\sum_{m\geq 1} (u'^{-1}-1)^m x^m = \frac{(\frac{x}{u'} - x)}{1 - \frac{x}{u'} + x}, \quad \sum_{n\geq 1} (1+x^{-1})^n u^n = \frac{(u + \frac{u}{x})}{1 - u - \frac{u}{x}},$$

$$g(u) = \int_x \frac{f(x)(u+\frac{u}{x})}{1-u-\frac{u}{x}} \frac{dx}{2\pi i x}, \quad f(x) = \int_{u'} \frac{g(u')(\frac{x}{u'} - x)}{1 - \frac{x}{u'} + x} \frac{du'}{2\pi i u'},$$

$$g(u) = \int_{u',x} \frac{g(u')(\frac{x}{u'} - x)(u + \frac{u}{x})}{(1 - \frac{x}{u'} + x)(1 - u - \frac{u}{x})} \frac{du'}{2\pi i u'} \frac{dx}{2\pi i x},$$

Can we verify our result using Fourier analysis and complex analysis method?

$$K(u, u') = \frac{1}{2\pi i} \int_x \frac{(\frac{1}{u'} - 1)(ux + u)dx}{((\frac{1}{u'} - 1)x - 1)((u-1)x + u)} = \frac{(\frac{1}{u'} - 1)u(\frac{u}{1-u} + 1)}{((\frac{1}{u'} - 1)\frac{u}{1-u} - 1)(u - 1)} = \frac{(1-u')u}{(u'-u)(1-u)},$$

In the above identity, I used residue theorem. For fixed $u, u' \in \mathbb{C}$, consider $x = re^{2\pi i\theta}$ such that $1 > |u'| > |x| > |u| > 0$, the only pole of the integrand in the region $|x| \leq r$ is $x = \frac{u}{1-u}$. So it suffices to verify that

$$g(u) = \frac{1}{2\pi i} \int_{u'} g(u') \frac{u(1-u')du'}{u'(u'-u)(1-u)},$$

Since $g(0) = 0$, the only pole of the integrand in the region $|u'| \leq r_0$ is $u' = u$, so the above identity holds by residue theorem.

An attempt to deduce Poisson kernel:

$$g(u') = \sum_{n\geq 1} b_n u'^n, \quad g(u) = \sum_{n\geq 1} b_n u^n, \quad |u'| = r_0, \quad |u| = r_1, \quad u' = r_0 e^{2\pi i\xi}, \quad u = r_1 e^{2\pi i\eta},$$

$$f_0(\xi) \triangleq g(u') = \sum_{n\geq 1} b_n r_0^n e^{2\pi i n\xi}, \quad b_n r_0^n = \int_{\mathbb{T}} f_0(\xi) e^{-2\pi i n\xi} d\xi,$$

$$f_1(\eta) \triangleq g(u) = \sum_{n\geq 1} \int_{\mathbb{T}} f_0(\xi) e^{-2\pi i n\xi} d\xi r_0^{-n} r_1^n e^{2\pi i n\eta} = \int_{\mathbb{T}} f_0(\xi) \sum_{n\geq 1} (\frac{r_1}{r_0})^n e^{2\pi i n(\eta-\xi)} d\xi,$$

$$\sum_{n\geq 1} (\frac{r_1}{r_0})^n e^{2\pi i n(\eta-\xi)} = \frac{\frac{r_1}{r_0} e^{2\pi i(\eta-\xi)}}{1 - \frac{r_1}{r_0} e^{2\pi i(\eta-\xi)}} = \frac{u}{u'-u}, \quad g(u) = \frac{1}{2\pi i} \int_{u'} g(u') \frac{u du'}{u'(u'-u)},$$

Problem 51 (1682 Crazy Professor)**.** Disjoint set. Implemented this struct with path compression and union by rank heuristics.

Problem 52 (1691 Algorithm Complexity)**.** Given a graph G with n vertices and m edges, which doesn't contain multiple arcs but may contain loops. Let $F(N)$ be the number of walks of length N from vertex n_s to vertex n_t, find the growth order of $F(N)$. Assume that the adjacency matrix of G

is A, then A is unsymmetric with 0-1 entries, $F(N) = A^N[n_s, n_t]$. Using the method of generating function, we get

$$\sum_{N\geq 0} A^N t^N = (I - At)^{-1}, \quad Answer = coeff < t^N, (I - At)^{-1}[n_s, n_t] >,$$

Consider the special case when G is a directed acyclic graph, and assume that vertices $v_1, v_2, ..., v_n$ are sorted topologically. Then A is a upper triangular 0-1 matrix. We may assume that $n_s = 1$ and $n_t = n$ by only considering the subgraph generated by vertices numbered from n_s to n_t. Now $F(N)$ reaches its maximum when $A[i, j] = 1$ for all $i \leq j$. In this scenario, we may write

$$A = I + B + B^2 + ... + B^{n-1}, \quad B[i, i+1] = 1, \quad 1 \leq i \leq n-1, \quad B^n = 0, \quad A = (I - B)^{-1},$$

Goal is to calculate the coefficient of B^{n-1} in the expansion of $A^N = (I - B)^{-N}$.

$$(I - B)^{-N} = \sum_{0\leq i\leq n-1} \binom{i+N-1}{i} B^i, \quad \binom{n+N-2}{n-1} = O(N^{n-1}),$$

Assume that the order of $F(N)$ is f, we know that f is at most $n-1$. How to determine its exact value? There are two cases depending on whether v_n can be reached from v_{n-1}. Assume $F_0(N)$ is the number of length N paths that don't pass by v_{n-1}, $F_1(N)$ is the number of paths that pass by v_{n-1}.

1) $A[n-1, n] = 0$, then no path from v_1 to v_n passes through v_{n-1}, we can remove this vertex and the number of vertices is reduced to $n-1$, $f = f_0$.

2) $A[n-1, n] = 1$, then any path from v_1 to v_n can be classified by whether it passes by v_{n-1} or not. $F_0(N)$ can be calculated from the subgraph excluding v_{n-1}. $f = \max(f_0, f_1)$.

i) First we consider the case when $A[n, n] = 1$. Assume that i is the last time that the path is at v_{n-1}, we have

$$F_1(N) = \sum_{i=1}^{N-1} F_1^0(i), \quad F_1^0(i) = \text{number of length } i \text{ paths from } v_1 \text{ to } v_{n-1}, \quad f_1 = f_1^0 + 1,$$

ii) $A[n, n] = 0$. Now we have $F_1(N) = F_1^0(N-1), f_1 = f_1^0$.

We can process the graph forwardly, and denote $f[i]$ the order of length i paths from v_1 to v_i. Then if $A[i, i] = 1$, we let $f[i] = f[i] + 1$; if $A[i, j] = 1, i < j$, we let $f[j] = \max(f[j], f[i])$.

Next let us consider the general case when G is an arbitrary directed graph. I used Tarjan's algorithm to calculate its strongly connected components and they form a new graph which is acyclic. How to add edges on the new graph? There may be more than one paths between two components, and a component may not have self-loop. Depth first search is used to sort the components topologically. For each component, if it has more edges than vertices, then we think it has more than one self-loops; if its edge number equals to its vertex number, we think it has exactly one self-loop; otherwise we think it has no self-loop. The order of paths from v_s to v_t is the same as the order of paths between their components. Moreover, if $A[i, i] \geq 2$ in the components graph, we let $f[i] = +\infty$.

In a directed graph, accessibility from vertex A to vertex B is a partial order: 1) if $A \to B, B \to C$, then $A \to C$; 2) $A \to A$; 3) if $A \to B, B \to A$, then $A \simeq B$. The equality holds in the sense of mutually accessible, that is, A, B are in the same strongly connected component. Mutually accessible is an equivalence relation:

$$A \sim A, \quad A \sim B \iff B \sim A, \quad A \sim B, B \sim C, \Rightarrow A \sim C,$$

Problem 53 (1697 Sniper Shot)**.** Method 1: projection onto plane $z = 0$.
Method 2: projection onto the plane spanned by AB and e_z.

Problem 54 (1716 Alternative Solution)**.** Read in two integers N, S, let $a = S - 2N$, $b = 3N - S$, output is $\frac{2ab+b}{N}$.

Problem 55 (1747 Kingdom Inspection)**.** Let the input integers be N, P, let $n = N - 1$, then

$$Ans = \sum_{i=0}^{n} (-1)^{n-i} \frac{(n+i)!}{2^i} \binom{n}{i},$$

Problem 56 (1758 Bald Spot Revisited 2)**.** Vertices are numbered from 1 to n, $2 \leq n \leq 50$. There exists an undirected edge between vertices i and j only if $i|j$ or $j|i$. Find the longest path such that each vertex are visited at most once. The path must start from vertex 1. Reference: https://people.csail.mit.edu/virgi/6.s078/lecture17.pdf. The most significant speed improvement follows Shen Yang's idea in discussion. Shen Yang mentioned LKH heuristic, but I don't know what it is exactly.

Problem 57 (1763 Expert Flea)**.** Calculate the number of Hamiltonian cycles in the circulant graph $C_n^{1,3}$. The answers a_n satisfy the following linear recurrence with $k = 19$:

$$a_{n+k} = \sum_{i=1}^{k} c_i a_{n+k-i}, \quad n \geq 2, \quad [a_2, a_3, a_4, a_5, a_6, ...] = [16, 2, 32, 24, 58, 46, 144, 110, 312, ...],$$

$$[c_1, ..., c_k] = [1, 2, -1, 1, -2, -1, -2, -3, 3, -1, 5, 2, 2, 1, -2, 0, -3, 0, -1],$$

Two methods can be used to determine the values of k and $c_i, 1 \leq i \leq k$. One way is to use reduction modulo a big prime, calculate determinants of size $(k+1) * (k+1)$ matrices and solve linear systems as follows. The other is to solve linear system directly using numpy.

$$\det \begin{pmatrix} a_{n+k} & ... & a_n \\ & ... & \\ a_n & ... & a_{n-k} \end{pmatrix} = 0, \quad \begin{pmatrix} a_{n+k-1} & ... & a_n \\ & ... & \\ a_n & ... & a_{n-k+1} \end{pmatrix} \begin{pmatrix} c_1 \\ ... \\ c_k \end{pmatrix} = \begin{pmatrix} a_{n+k} \\ ... \\ a_{n+1} \end{pmatrix},$$

$a_6 = 58$ takes into acount of duplicated edges and satisfies linear recurrence, its correct output should be $a_6' = 12$. The values of a_2, a_3, a_4, a_5 have combinatorial explanations since we can construct circulant graphs $C_2^{1,3}, C_3^{1,3}, C_4^{1,3}, C_5^{1,3}$ explicitly.

Problem 58 (1768 Circular Strings)**.** First we have to check the orientation of the polygon, if the vertices are located in counter-clockwise order, then their order should be reversed. The criterion is to check whether $p[i+1] - p[i] = (p[i] - p[i-1]).rotate(\frac{2\pi}{N})$ is satisfied for each i.

Problem 59 (1797 Summit Online Judge Version 2)**.** Given $x, y, l, r \in [1, 10^{18}], l \leq r$, find the number of integers that can be written in the form of $ax + by, a, b \in \mathbb{N}$. First we assume that $\gcd(x, y) = 1$. It suffices to consider the case when $r \leq xy - x - y$, since all the integers no less than $xy - x - y + 1$ can be written in the above form.

$$f(r) = |\{ax + by \leq r, a, b \geq 0\}|, \quad f(r) = \sum_{0 \leq a \leq \lfloor \frac{r}{x} \rfloor} (\lfloor \frac{r - ax}{y} \rfloor + 1) = \sum_{0 \leq b \leq \lfloor \frac{r}{y} \rfloor} (\lfloor \frac{r - by}{x} \rfloor + 1),$$

We may assume that $x > y$. If $y = 1$, the answer is calculated directly as

$$\sum_{0\le a\le a_0}((r-ax)+1) = (r+1)(a_0+1) - \frac{a_0(a_0+1)x}{2},$$

Otherwise let $x = uy + x_1, 1 \le x_1 \le y-1, a_0 = \lfloor \frac{r}{x} \rfloor$,

$$\frac{r-ax}{y} = \frac{r - a_0uy + (a_0-a)uy - ax_1}{y}, \quad r_1 = r - a_0uy, \quad \lfloor \frac{r-ax}{y} \rfloor = \lfloor \frac{r_1 - ax_1}{y} \rfloor + (a_0-a)u,$$

The above step can be regarded as a change of slope. Next let

$$\lfloor \frac{r_1 - a_0x_1}{y} \rfloor = v, \quad r_1 = vy + r_2, \quad \lfloor \frac{r_1 - ax_1}{y} \rfloor = \lfloor \frac{r_2 - ax_1}{y} \rfloor + v,$$

This step can be thought as a translation. We have

$$\sum_{0\le a\le a_0}(\lfloor \frac{r-ax}{y} \rfloor + 1) = \sum_{0\le a\le a_0}(\lfloor \frac{r_1-ax_1}{y} \rfloor + 1 + (a_0-a)u) = \sum_{0\le a\le a_0}(\lfloor \frac{r_2-ax_1}{y} \rfloor + 1 + (a_0-a)u + v)$$
$$= \frac{(a_0+1)a_0u}{2} + v(a_0+1) + a_0 - a_2 + \sum_{0\le a\le a_2}(\lfloor \frac{r_2-ax_1}{y} \rfloor + 1),$$

The last equality is because

$$0 \le r_2 - a_0x_1 = r_1 - vy - a_0x_1 \le y-1, \quad a_2 = \lfloor \frac{r_2}{x_1} \rfloor \ge a_0,$$

Problem 60 (1798 Fire Circle Version 2)**.** Gauss circle problem.

Problem 61 (1807 Cartridges for Maxim)**.** Reference: *acm.timus.ru/forum/thread.aspx?id=25649.* Dynamic programming. Given a prime p, find integers $a_1, a_2, ..., a_k$ such that $p = \sum_{i=1}^k a_i$ and $\mathrm{lcm}(a_1, ..., a_k)$ attains its maximum, $k \ge 2$. p is the minimal prime divisor of n in problem statement. State transition equation is

$$dp[0] = 0, \quad dp[x] = -\infty, \quad x \ge 1, \quad dp[m] = \max\{dp[m], \log p^i + dp[m-p^i]\},$$

Notice that only the first $M = 114$ primes are used, so we can construct an $m*(p+1)$ array *decision* to store all the decision points of the equation above.

Problem 62 (1810 Antiequations)**.**

$$A : \mathbb{F}_3^n \to \mathbb{F}_3^k, \quad P_i = \{y_i = b_i\} \subset \mathbb{F}_3^k, \quad (y_1, ..., y_k) \in \mathbb{F}_3^k,$$

Assume that $l = im(A), \dim(l) = p$. We consider the case when all P_i cross intersects l first, let $l \cap P_i = Q_i$, then $\dim(Q_i) = p - 1$. It suffices to calculate the size of $l \setminus \bigcup_{1\le i\le k} Q_i$. For each pair of $i, j \in \{1, ..., k\}$, there are three possible relations between Q_i and Q_j: 1) $Q_i = Q_j$ identical, 2) $Q_i \cap Q_j = \emptyset$ parallel, 3) $\dim(Q_i \cap Q_j) = p-2$ cross intersect.

Method 2: counting points on the affine variety $X = Q_1 \cup Q_2 ... \cup Q_k$. Generally speaking, let

$$N_m = |X(\mathbb{F}_{q^m})|, \quad Z(X,t) = \exp(\sum_{m\ge 1} \frac{N_m}{m} t^m),$$

For example, when $p = k$, $Q_i : y_i = b_i$, we have

$$N_m = q^{mk} - (q^m - 1)^k = \sum_{i=0}^{k-1} q^{mi} \binom{k}{i} (-1)^{k+i+1},$$

$$\sum_{m\geq 1} \frac{N_m}{m} t^m = \sum_{i=0}^{k-1} \binom{k}{i} (-1)^{k+i+1} \sum_{m\geq 1} \frac{q^{mi}}{m} t^m = \sum_{i=0}^{k-1} \binom{k}{i} (-1)^{k+i} \log(1 - q^i t),$$

$$Z(X,t) = \exp(\sum_{i=0}^{k-1} \binom{k}{i} (-1)^{k+i} \log(1 - q^i t)) = \frac{(1-q^{k-2}t)^{\binom{k}{2}} \dots}{(1-q^{k-1}t)^{\binom{k}{1}} (1-q^{k-3}t)^{\binom{k}{3}} \dots},$$

$$N_1 = \frac{dZ}{dt}\Big|_{t=0} = \binom{k}{1} q^{k-1} - \binom{k}{2} q^{k-2} + \binom{k}{3} q^{k-3} \dots = q^k - (q-1)^k,$$

Another special case is when $b_i = 0$, each P_i is a codimension 1 subspace of $\mathbb{F}_3^k$. Claim: we may select a basis $s_1, ..., s_p$ of l such that their supports are pairwise disjoint.

Problem 63 (1812 The Island of Bad Luck 2)**.** Given integers r_1, r_2, N, n,

$$\sqrt{(r_1 + r)^2 - (r_1 - r)^2} + \sqrt{(r_2 + r)^2 - (r_2 - r)^2} = \sqrt{(r_1 + r_2)^2 - (r_1 - r_2)^2},$$

$$2\sqrt{r_1 r} + 2\sqrt{r_2 r} = 2\sqrt{r_1 r_2}, \quad \sqrt{r} = \frac{\sqrt{r_1 r_2}}{\sqrt{r_1} + \sqrt{r_2}},$$

1) $n = 0$ or $n = 2^N$, output r_1 or r_2.
2) $r_1 r_2$ is not a perfect square, output "Irrational".
3) $\gcd(r_1, r_2) = d$, $r_1 = r_1' d$, $r_2 = r_2' d$, calculate r recursively. Dynamic programming is used to store calculated values of r_n in a table.

Problem 64 (1813 Random Shuffler)**.** Wrote a python program to verify Lingxi Xie's hint. The answer for input M is $\lfloor \frac{1+M+1-d}{2} \rfloor$ where d is the calculated common difference of the arithmetic sequence. Reference: *acm.timus.ru/forum/?space=1&num=1813*.

Problem 65 (1814 Continued Fraction)**.** We need to implement quadratic extension of rational number field as a struct QuadraticRT. An element's inverse is given by

$$(\frac{x + y\sqrt{N}}{z})^{-1} = \frac{z(y\sqrt{N} - x)}{Ny^2 - x^2},$$

$nums[i]$ stores quadrtic rational $a_i + r_i$, where $a_i \in \mathbb{Z}_+, 0 < r_i < 1$, and $\sqrt{N} = nums[0]$. Formula of continued fraction is given by $nums[i+1] = \frac{1}{r_i}$. By the following theorem, we may check that if the block length is m, then

$$nums[m+1] = nums[1], \quad nums[m] = a_0 + nums[0] = a_0 + \sqrt{N},$$

Method 1: assume that $R_n = \frac{P_n}{Q_n} = [a_0; a_1, a_2, ..., a_n]$ and $1 \leq k_1 \leq m, k_1 \equiv k \mod m$, then

$$P_n = P_{n-1} a_n + P_{n-2}, \quad Q_n = Q_{n-1} a_n + Q_{n-2},$$

We start by calculating $[a_{k_1+1}, ...a_k]$. Let $P^l = P_{lm}, Q^l = Q_{lm}, l = (k - k_1)/m$ in this scenario, then

$$\begin{pmatrix} P^l \\ Q^l \end{pmatrix} = \begin{pmatrix} P_m & P_{m-1} \\ Q_m & Q_{m-1} \end{pmatrix} \begin{pmatrix} P^{l-1} \\ Q^{l-1} \end{pmatrix} = M^l \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

Turning back to the original problem, we have

$$P_k = P^l P_{k_1} + Q^l P_{k_1-1}, \quad Q_k = P^l Q_{k_1} + Q^l Q_{k_1-1},$$

Method 2: define $P_{-1} = 1, Q_{-1} = 0, P_{-2} = 0, Q_{-2} = 1$,

$$A_i = \begin{pmatrix} a_i & 1 \\ 1 & 0 \end{pmatrix}, \quad M = A_m A_{m-1} ... A_1, \quad prodm = A_{k_1} ... A_1 M^l A_0,$$

Theorem 3. If $r \in \mathbb{Q}, r > 1$ is not a perfect square, then

$$\sqrt{r} = [a_0; \overline{a_1, a_2, ..., a_2, a_1, 2a_0}],$$

It has a repeating block of length m, in which the first $m-1$ partiail denominators form a palindromic string. In the continued fraction expansion of $\frac{P+\sqrt{D}}{Q}$, the largest partial denominator a_i in the expansion of $\sqrt{D}$ is less than $2\sqrt{D}$, and the block length $m = L(D)$ is less than $2D$. A sharper bound is

$$L(D) = O(\sqrt{D} \log D),$$

Problem 66 (1815 Farm in San Andreas)**.** Given coordinates of A, B, C, costs c_A, c_B, c_C, find the minimum of $c_A PA + c_B PB + c_C PC$. If P is different from A, B, C, assume that $\alpha = \angle BPC, \beta = \angle CPA, \gamma = \angle APB$, $\alpha_1 = \pi - \alpha, \beta_1 = \pi - \beta, \gamma_1 = \pi - \gamma$, then

$$c_B = c_A \cos \gamma_1 + c_C \cos \alpha_1, \quad c_A = c_C \cos \beta_1 + c_B \cos \gamma_1, \quad c_C = c_B \cos \alpha_1 + c_A \cos \beta_1,$$

$\alpha_1, \beta_1, \gamma_1$ are interior angles of the triangle *coststri* with edge lengths c_A, c_B, c_C. Geometrically we may construct point R (temppt in program) such that $BC : CR : BR = c_A : c_B : c_C$. Intersection of AR with the circumcircle of $\triangle CBR$ is the point P required.

Problem 67 (1816 Troubles with Pollard)**.**

Problem 68 (1840 Victim of Advertising)**.** The trajectory of the skater is uniquely determined since only one segment can be extended if there is no arc connecting two consecutive directed segments without breaks. There are three constraints:

$$v \le 10\text{m/s}, \quad a_{tan} \le 1\text{m/s}^2, \quad a_n \le 1\text{m/s}^2,$$

My approach is to draw a $\frac{v^2}{2} - s$ diagram, $\frac{v^2}{2}$ is the kinetic energy.

$$a_{tan} = \frac{dv^2}{2ds} \le 1, \quad a_n = \frac{v^2}{R} \le 1, \quad \frac{v^2}{2} \le \frac{R}{2},$$

In this motion planning problem, the trajectory consists of N segments and $N-1$ arcs, separated by $2N$ endpoints. Assume that the kinetic energy and traveled length at the i-th endpoint are $kinetic[i]$ and $s[i]$, then constraints are

$$|kinetic[i] - kinetic[i+1]| \le s[i+1] - s[i], \quad kinetic[2i+1], kinetic[2i+2] \le \min(\frac{arcs[i].radius}{2}, 50),$$

Notice that if solution 1 and 2 are legal, then their pointwise maximum is also legal. So we may regard it as a linear programming problem that maximizes $\sum_{1 \le i \le 2N-1} kinetic[i]$. We may solve this problem in a way similar to Ford-Fulkerson algorithm by finding possible increments of $kinetic[i]$ in every iteration. Condition that $kinetic[i]$ can't increase is that at least one of the three constraints at endpoint i holds.

Thus we get $kinetic[i]$ exactly from the greedy algorithm above. trapzoidtime() is used to calculate for costed time in trapzoidal regions of the $k - s$ diagram, and elapsedtime() calculates costed time between two consecutive endpoints. I got memory limit exceeded sevaral times, possibly because call elapsedtime() recursively. Changing parameter types from T to constant reference doesn't cost much memory when T is small.

Problem 69 (1845 Integer-valued Complex Determinant)**.** Calculate the determinant of a Gaussian integer valued matrix. Attention: my implementation of struct BigInteger is written in this program. It is modified from Qifeng Chen's implementation. Const qualifiers are used as improvements, and BigNumber uses std::vector instead of array to store x.

Method 1: Integer coefficient Gaussian elimination. GaussianZT is implemented as Gaussian integer struct. Use extended Euclidean algorithm to calculate GCD of Gaussian integers after pivoting, so that all the elementary row transformations have Gaussian integer coefficients. A prototype of it was implemented in IntegerElimination.cpp, in which I only tested on integer coefficient matrices but not Gaussian integer. There are three kinds of elementary row transformations:

$$a_j \to a_j - ca_i, \quad a_j \to -a_j, \quad a_i \leftrightarrow a_j,$$

and I try to turn a_{kk} into $\gcd(a_{kk}, a_{ik})$ for $i > k, a_{ik} \neq 0$ after pivoting. Let $u = a_{kk}, v = a_{ik}$, the output of extendedGCD function satisfies

$$xu + yv = d, \quad u = du_q, v = d_v1, \quad xu_1 + yv_1 = 1, \quad u_1v - v_1u = 0, \quad |x| < |v_1|, |y| < |u_1|,$$

So we can take elementary row transform on the k, i-th rows resulting in

$$\det \begin{pmatrix} x & y \\ -v_1 & u_1 \end{pmatrix} = 1, \quad \begin{pmatrix} x & y \\ -v_1 & u_1 \end{pmatrix} \begin{pmatrix} a_{kk} \\ a_{ik} \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix},$$

and it doesn't change the determinant of A. Recursion of extendedeGCD satisfies

$$u\%v = u - \lambda v, \quad d = yv + x(u\%v) = xu + (y - \lambda x)v,$$

Note that generally speaking, we don't need auxilliary matrices P, L that appears in $PA = LU$ while calculating determinant.

Method 2: GaussianQT is implemented as Gaussian rational struct. Use raw Gaussian elimination after pivoting, but the current result is RTE15 while using BigInteger struct, WA15 while using long long.

Problem 70 (1950 Martian Farm)**.** Bisect until F is on an altitude or the size of the new farmlands is negligibly small. During every query, we know F is strictly inside isosceles right triangle ABC, and we ask how much area of $\triangle ABC$ is selected. Suppose D is the midpoint of AB, then exactly one of the following two cases happens:

1) OF only intersects with one of the two smaller triangles;

2) OF intersects with both the two smaller triangles, then OF cross intersects CD.

It suffices to check if OF cross intersects CD, and judge if F is strictly inside $\triangle ACD, \triangle BCD$ or on edge CD. Use long double to improve precision.

Problem 71 (1951 Complex Root)**.** Given $a = a_x + ia_y, b = b_x + ib_y$ and integers n, m, find the number of complex solutions to $x^n = a, x^m = b$. Method is to use modified Euclidean algorithm to find $d = \gcd(n, m)$ and x^d.

Problem 72 (1953 Biggest Inscribed Ellipse)**.** It suffices to consider the case when the triangle Γ is an equilateral Γ_0 with edge length $2\sqrt{3}$, hence its inscribed circle has radius 1.

Method 1: Calculate the transformation from Γ to Γ_0. Assume the transformation matrix is A, then the inscribed circle is

$$(x\ y)A^tA\begin{pmatrix}x\\y\end{pmatrix}=1,\quad B=A^tA,\quad \lambda^2-\mathrm{tr}(B)\lambda+\det(B)=0,$$

$$\lambda_1<\lambda_2,\quad a^2=\frac{1}{\lambda_1},\quad b^2=\frac{1}{\lambda_2},\quad c^2=\frac{1}{\lambda_1}-\frac{1}{\lambda_2},$$

Method 2: Calculate the transformation from Γ_0 to Γ. Assume the transformation matrix is A, a^2,b^2 satisfy the following equation:

$$B=\mathrm{tr}(AA^t),\quad C=\det(AA^t),\quad r^2-Br+C=0,$$

I got WA-47 several times at first, trick to pass this testcase is to normalize the triangle perimeter to 3 and multiply the scale to the answer before output.

Problem 73 (1973 Location Generator)**.** Wrote a python program to estimate its value on a discrete grid but later I found it unnecessary. One dimensional analogue: let $b,c\sim U([0,1])$, then

$$\mathbb{E}|b-c|=\int_{[0,1]^2}|b-c|=\frac{1}{3},$$

Two dimensional case: 1) Given two disjoint triangles T_1,T_2 with a common point $A=(0,0)$. $B\in T_1, C\in T_2$ satisfy uniform distributions and assume that $AB\times AC\geq 0$, then

$$\mathbb{E}S_{\triangle ABC}=\mathbb{E}\frac{AB\times AC}{2}=\mathbb{E}\frac{x_By_C-x_Cy_B}{2}=\frac{\overline{x_By_C}-\overline{x_Cy_B}}{2},$$

2) $B,C\in T_1$ are in the same triangle. Since the expectation varies bilinearly under plane affine transformation, we may assume that $P(1,0),Q(0,1),T_1=\triangle APQ$. Let $AB\cap PQ=D$, then

$$\mathbb{E}S_{\triangle ABC}=\int_{T_1}dB\Big(\frac{x_B\overline{y_{C,l}}-\overline{x_{C,l}}y_B}{2}x_D+\frac{-x_B\overline{y_{C,r}}+\overline{x_{C,r}}y_B}{2}y_D\Big),$$

$$\overline{x_{C,l}}=\frac{x_D}{3},\quad \overline{y_{C,l}}=\frac{1+y_D}{3},\quad C\in T_1^l,\quad \overline{x_{C,r}}=\frac{1+x_D}{3},\quad \overline{y_{C,r}}=\frac{y_D}{3},\quad C\in T_1^r,$$

$$\mathbb{E}S_{\triangle ABC}=\int_{T_1}dB\Big(\frac{x_B\frac{1+y_D}{3}-\frac{x_D}{3}y_B}{2}x_D+\frac{-x_B\frac{y_D}{3}+\frac{1+x_D}{3}y_B}{2}y_D\Big)=\int_{T_1}dB\Big(\frac{x_B}{6}x_D+\frac{y_D}{6}y_D\Big)$$
$$=\int_{[0,1]}dx_D\frac{x_D^2+y_D^2}{9}=\int_0^1\frac{2x^2-2x+1}{9}dx=\frac{2}{27},$$

we used the fact that when D is the same, $\overline{x_B}=\frac{2x_D}{3},\overline{y_B}=\frac{2y_D}{3}$.

Problem 74 (1975 Model of the Earth)**.** Given n points on the unit sphere, calculate the number of rotations that is a permutation of itself. Current status: WA-25. For any two different points A,B, if $A\neq -B$, then there exists an unique rotation $\varphi_{AB}:A\mapsto(0,0,1),\varphi(B)\in Oxz$. Finetune around a solution candidate: assume that $roll=\alpha, pitch=\beta, yaw=\gamma$, $Rq_i\sim p_i, R\in SO(3)$,

$$\begin{pmatrix}1&-\gamma&\beta\\\gamma&1&-\alpha\\-\beta&\alpha&1\end{pmatrix}(q_1,q_2,...,q_n)\sim(p_1,p_2,...,p_n),$$

$$\begin{pmatrix} 0 & q_i.z & -q_i.y \\ -q_i.z & 0 & q_i.x \\ q_i.y & -q_i.x & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \sim \begin{pmatrix} p_i.x - q_i.x \\ p_i.y - q_i.y \\ p_i.z - q_i.z \end{pmatrix}$$

Find least square solution to $3n$ linear equations in 3 variables α, β, γ.

Problem 75 (1983 Nectar Gathering)**.** Calculate the area of ball triangle intersection in 3 dimensional Euclidean space. Step 1: projection onto the triangle plane. Step 2: calculate the intersections of the circle with each of the 3 triangle edges.

Problem 76 (1996 Cipher Message 3)**.** Input: n is the size of image file in bytes, m is the size of secret information in bytes. Let a_i be the last bit of the i-th image, b_j be the last bit of the j-th information, c_i be the first 7 bits of the i-th image, d_j be the first 7 bits of the j-th information. Wrote kmpmatcher() function to check all the occurences of d in c using KMP algorithm. Use FFT and prefix sum to calculate $e_l = \sum_{j=0}^{m-1} b_j + a_{j+l} - 2b_j a_{j+l}$, which is the distance between image and shifted information.

Problem 77 (2038 Minimum Vertex Cover)**.** Given a bipartite graph without duplicated edges. For each vertex, determine whether it is contained in any minimum vertex cover of the graph. Claim: let F be the maximum matching number, left and right partial derivatives $\frac{\partial F}{\partial c_i}$ of F can only have three possibilities: $(0,0),(1,0),(1,1)$. Moreover, for a given maximum matching, sum of derivatives at matched vertices always equals 2. Actually these partial derivatives can be calculated by running depth first search on the residual network twice. My implementation based on Dinic maxflow got TLE-25 or TLE-26. Wrote Hopcroft-Karp bipartite matching using cpp for the first time and got accepted after using readint() instead of std::cin. Notice that Hopcroft-Karp algorithm uses breadth first search and queue struct.

Problem 78 (2061 OEIS A216264)**.** 1) Length n string has at most n different palindromic substrings, in which case we say the string is rich.

2) If S is rich, c is a symbol, then Sc has at most one different palindromic substring, which is the longest palindromic suffix of Sc.

3) Prefix of a rich string is rich. So all substrings of a rich string are rich.

To solve the original problem, we may first generate all the rich strings of length 60. Notice that there are certain inclusion relations between palindromic strings: length n palindromic string contains palindromic substrings of length $n-2, n-4, \ldots$.

Problem 79 (2076 Vasiana)**.**

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ d_0 & e_0 & f_0 \end{pmatrix} \begin{pmatrix} t^2 \\ t \\ 1 \end{pmatrix}, \quad \begin{pmatrix} t^2 \\ t \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix},$$

1) $x, y, 1$ are linearly independent. The trajectory of (x,y) must be a parabola.

$$ax + by + c = (dx + ey + f)^2, \quad x' = \frac{dx + ey}{\sqrt{d^2 + e^2}}, \quad y' = \frac{-ex + dy}{\sqrt{d^2 + e^2}},$$

$$T = \frac{1}{\sqrt{d^2 + e^2}} \begin{pmatrix} d & e \\ -e & d \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} = T^t \begin{pmatrix} x' \\ y' \end{pmatrix},$$

$$\begin{pmatrix} t^2 \\ t \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} T^t & \\ & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a' & b' & c \\ d' & 0 & f \end{pmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}, \quad d' = \sqrt{d^2 + e^2},$$

$$(d'x'+f)^2 = a'x'+b'y'+c, \quad y' = a''(x'-x'_{mid})^2 + c'',$$

$$x'_{mid} = \frac{-f+\frac{a'}{2d'}}{d'}, \quad a'' = \frac{d'^2}{b'}, \quad c'' = \frac{f^2-c}{b'},$$

The equation for circle parabola intersection is

$$x^2 + (y-y_0)^2 = R^2, \quad y = a''x^2 + c'',$$

$$f(x^2) = a''^2x^4 + (2a''(c''-y_0)+1)x^2 + (c''-y_0)^2 - R^2 = 0,$$

Criterion for existence of intersection is, f has non-negative real root.

2) $x, y, 1$ are linearly dependent. i) $a_0 = d_0 = 0$, a) $b_0 = e_0 = 0$, it means that the trajectory of (x, y) is a single point. b) $b_0 \neq 0$ or $e_0 \neq 0$, it means that the trajectory of (x, y) is a full straight line. ii)$a_0 \neq 0$ or $d_0 \neq 0$, now the trajectory of (x, y) is a ray.

$$P_{critical}(x_{critical}, y_{critical}), \quad x' = \frac{a_0x+d_0y}{\sqrt{a_0^2+d_0^2}}, \quad y' = \frac{-d_0x+a_0y}{\sqrt{a_0^2+d_0^2}},$$

Problem 80 (2083 The Guardian of Traditions)**.** $p : [n] \to [n]$ is a permutation, $\{q_i\}_{i=1}^m$ is a integer vector. Calculate $\sum_{i=1}^m q_i p^i(j)$ for each $1 \le j \le n$.

Fast Fourier Transform. Use std::llround to round a double variable to long long, use std::lround to round a double variable to long. Naive multiplication algorithm gets TLE-9. Accepted algorithm first calculates the loop decomposition of the permutation. Vector q is divided into 4 parts to improve precision. Finally, length of vector q is aligned to the loop length before each FFT multiplication.

Problem 81 (2085 Magic Programmer)**.** Find a path on a tree such that Ivan can try each of the m technologies exactly once. Key idea is to use centroid decomposition. For a given root vertex, all the paths from v_i to v_j can be classified into 2 classes: 1) paths that pass through the root vertex; 2) paths that don't pass through the root vertex, such a path must be totally contained in a single subtree of the root vertex. So we may design an algorithm to solve the problem recursively by carefully selecting the root vertex for each subtree. While processing a subtree, we only consider paths that pass through the root vertex and paths whose length is at least 2. For each path, a tuple $(x_1, x_2, ..., x_m) \in \mathbb{Z}^m$ is associated to it indicating that the path has x_i copies of the i-th technology. For any vertex v, $tech(v) \in \mathbb{Z}^m$ with $0, 1$ entries indicates the set of technologies associated to this vertex. Notice that $\mathbb{Z}^m$ is an additive abelian group. For a given subtree and a given root vertex, let $set(v_i) = \sum_{p \in path(v_i, root)} tech(p)$. The condition that the path from v_i to v_j satisfies the requirement is given by $set(v_i) + set(v_j) - tech(root) = [m]$. We only need to consider the case when $x_i = 0, 1$ for $set(v) = (x_1, x_2, ..., x_m)$ since if the path from v_i to v_j satisfies the requirement and pass through $root$, then each technology appears at most once in $set(v_i)$ and $set(v_j)$.

Special case when the tree is a chain: suppose that the vertices are $v_1, v_2, ..., v_n$ and the edges are (v_i, v_{i+1}), $set(v_i) = \sum_{j=1}^i tech(v_j)$. Then the requirement is

$$set(v_j) - set(v_i) + tech(v_i) = set(v_j) - set(v_{i-1}) = [m], \quad i \le j,$$

We may scan the chain in order, save $[m] + set(v_{i-1})$ in a dictionary, and query $set(v_j)$. Preprocessing $set(v_i)$ takes linear time.

The second key point is to construct a hash function

$$\varphi : \mathbb{Z}^m \to \mathbb{Z}/modulo\mathbb{Z}, \quad (x_1, x_2, ..., x_m) \mapsto \sum_{i=1}^m x_i p^{i-1},$$

where *modulo* is a prime and p is a primitive root in $\mathbb{Z}/modulo\mathbb{Z}$. It is an additive homomorphism. I wrote *primitivetest.py* to calculate primitive roots modulo a prime P. It also calculates the factorization of $P-1$, and contains an implementation of linear sieve.

In order to test my code, I wrote a random tree generator. I wrote a function to count the number of paths in the tree with any given length, and I wrote a checker program.

Reference: *en.wikipedia.org/wiki/Prüfer_sequence.*

Problem 82 (2099 Space Invader)**.** Four conditions:

$$AB \cdot CD = 0, \quad AB \cdot (BC \times CD) = 0, \quad AB \cdot BC \geq 0, \quad CD \cdot BC \geq 0,$$

Problem 83 (2117 Polyphemus' triples)**.**

$$\sqrt{A} + \sqrt{B} = \sqrt{C}, \quad A = a^2 D, \quad B = b^2 D, \quad C = c^2 D, \quad D \text{ is square-free.}$$

Then the answer is $\lceil \frac{c+1}{2} \rceil$. Our goal is to calculate the square part c^2 of C. Since $0 \leq C \leq 10^{18} < 2^{60}$, it suffices to calculate the list of primes up to 2^{20}. If the remaining integer n doesn't have any prime factor less than 2^{20}, and want to know its square part. 1) $n = n_0^2$ and n_0 is a prime. 2) $n = n_1$, n_1 is square-free and has at most 2 prime factors. 3) It is impossible that $n = n_0^2 n_1$ where $n_0, n_1 > 1$, n_1 is square-free.

Problem 84 (2121 Intersection of parabolas)**.** Find the area of the region bounded by the following two parabolas: $y = (x-a)^2$ and $x = (y-a)^2$, $1 \leq a \leq 10^{18}$ is an integer.

The region is a curved quadrilateral, and we want to know the coordinates of its four vertices $V_i(x_i, y_i), 1 \leq i \leq 4$. The coordinates of two vertices on the line $x = y$ are the roots of $x = (x-a)^2$,

$$x^2 - (2a+1)x + a^2 = 0, \quad x_1 = y_1 = \frac{2a+1-\sqrt{4a+1}}{2}, \quad x_2 = y_2 = \frac{2a+1+\sqrt{4a+1}}{2},$$

So these two vertices are $V_1(x_1, x_1), V_2(x_2, x_2)$. Using the substitution $t = a - y$, we have

$$t^2 = x = (y-a)^2, \quad 0 = x^2 - 2ax + a^2 - y = t^4 - 2at^2 + t + a^2 - a = P(t),$$

We've already know that its two roots are

$$t_1 = \frac{-1+\sqrt{4a+1}}{2}, \quad t_2 = \frac{-1-\sqrt{4a+1}}{2}, \quad t_1^2 = x_1, \quad t_2^2 = x_2,$$

Denote $Q(t) = t^4 - (2a+1)t^2 + a^2$, we have the following factorization of $P(t)$:

$$P(t) - Q(t) = t^2 + t - a = (t-t_1)(t-t_2), \quad P(t) = (t^2+t-a)(t^2-t-a+1),$$

So the remaining two roots of $P(t)$ are

$$t_3 = \frac{1+\sqrt{4a-3}}{2}, \quad t_4 = \frac{1-\sqrt{4a-3}}{2},$$

Their corresponding coordinates are

$$x_3 = y_4 = \frac{2a-1+\sqrt{4a-3}}{2}, \quad x_4 = y_3 = \frac{2a-1-\sqrt{4a-3}}{2},$$

The desired area is

$$Area = (x_3 - x_1)^2 + 2\int_{y_3}^{y_1} (x_3 - (a-y)^2)dy + 2\int_{x_3}^{x_2} (x - (a-x)^2)dx = I + II + III,$$

$$\frac{II}{2} = (x_3 - a^2)(y_1 - y_3) + a(y_1^2 - y_3^2) - \frac{y_1^3 - y_3^3}{3},$$

$$\frac{III}{2} = -a^2(x_2 - x_3) + (a + \frac{1}{2})(x_2^2 - x_3^2) - \frac{x_2^3 - x_3^3}{3},$$

Numerical results show that actually we have $Area = 4a - \frac{1}{3}$. Its rigorous proof through algebraic computation is as follows:

$$x_1, x_2 \text{ satisfy: } x^2 - (2a+1)x + a^2 = 0,$$

$$x_3, x_4 \text{ satisfy: } x^2 - (2a-1)x + (a-1)^2 = 0,$$

$$\frac{II + III}{2} = x_3(x_1 - x_4) + \frac{x_2^2 - x_3^2}{2} - a^2(x_1 + x_2 - x_3 - x_4) + a(x_1^2 + x_2^2 - x_3^2 - x_4^2) - \frac{x_1^3 + x_2^3 - x_3^3 - x_4^3}{3},$$

Let $IV = x_3(x_1 - x_4) + \frac{x_2^2 - x_3^2}{2}$, the remaining terms can be calculated through Vieta's theorem.

$$x_1 + x_2 - x_3 - x_4 = 2a + 1 - (2a - 1) = 2,$$

$$x_1^2 + x_2^2 - x_3^2 - x_4^2 = (2a+1)^2 - 2a^2 - (2a-1)^2 + 2(a-1)^2 = 4a + 2,$$

$$x_1^3 + x_2^3 - x_3^3 - x_4^3 = (2a+1)((2a+1)^2 - 3a^2) - (2a-1)((2a-1)^2 - 3(a-1)^2) = 6a^2 + 12a - 1,$$

$$\frac{II + III}{2} = IV - 2a^2 + a(4a+2) - \frac{6a^2 + 12a - 1}{3} = IV - 2a + \frac{1}{3},$$

$$Area = (x_3 - x_1)^2 + 2 \cdot IV - 4a + \frac{2}{3} = x_1^2 + x_2^2 - 2x_3x_4 - 4a + \frac{2}{3}$$

$$= (2a+1)^2 - 2a^2 - 2(a-1)^2 - 4a + \frac{2}{3} = 4a - \frac{1}{3},$$

Problem 85 (2126 Partition into teams)**.** Method 1: raw search that iterates through all possible patterns, received TLE-3. Method 2: dynamic programming, $dp[k]$ is the number of possible states that $countred - countblue = k$ after n steps, received TLE-7. Method 3: modelling as one dimensional random walk, received TLE-12.

$$Answer = \frac{1}{2}(3^N - \sum_{a+2b=N} \binom{N}{a, b, b}),$$

Method 4: making use of modulo P arithmetic. How to calculate $\binom{N}{a,b,b} \mod P$?

$$(x+y+z)^P = x^P + y^P + z^P, \quad (x+y+z)^{P^2} = x^{P^2} + y^{P^2} + z^{P^2}, \quad (x+y+z)^{P^n} = x^{P^n} + y^{P^n} + z^{P^n},$$

$$N = \sum_{i=0}^{k} a_i P^i, \quad 0 \le a_i \le P - 1, \quad a_k \ne 0,$$

$$\binom{N}{a, b, b} \mod P = coeff < x^a y^b z^b, (x+y+z)^N > \mod P,$$

$$(x+y+z)^N = \prod_{i=0}^{k} (x^{P^i} + y^{P^i} + z^{P^i})^{a_i} \mod P,$$

$$\sum_{a+2b=N} \binom{N}{a,b,b} = coeff < 1, (1+x+x^{-1})^N >= coeff < 1, \prod_{i=0}^{k} (1+x^{P^i}+x^{-P^i})^{a_i} >$$

$$= \prod_{i=0}^{k} coeff < 1, (1+x^{P^i}+x^{-P^i})^{a_i} > \mod P,$$

$$\sum_{i=1}^{P-1} i^k = 0 \mod P, \quad 1 \le k \le P-2, \quad \sum_{i=1}^{P-1} i^{P-1} = -1 \mod P,$$

$$(1+x+x^{-1})^k = \sum_{j=-k}^{k} c_j x^j, \quad (P-1)c_0 = \sum_{x=1}^{P-1} (1+x+x^{-1})^k \mod P, \quad k < P-1,$$

$$(P-1)(1+c_0+1) = \sum_{x=1}^{P-1} (1+x+x^{-1})^{P-1} \mod P,$$

So we may calculate $c_0 = coeff < 1, (1+x+x^{-1})^k > \mod P$ in $O(P \log k)$ time. The range $N \le 10^{18}$ and $5 \le P < 10^6$ implies that the base P expansion of N doesn't have much digits, and I got accepted using c++.

Problem 86 (2150 4B and Zoo)**.** By the following theorem of Dirac, G has a Hamiltonian cycle since $d(v) \ge \lceil \frac{n}{2} \rceil$ for each vertex $v \in G$. When n is even, this cycle gives an perfect matching as desired.

Theorem 4 (Dirac, 1952)**.** A simple graph with $n \ge 3$ vertices contains a Hamiltonian cycle if every vertex has degree $\frac{n}{2}$ or greater.

Problem 87 (2151 Mahjong)**.** Split the tiles according their suits.

Problem 88 (2157 Skydiving)**.** Dynamic programming. Raw implementation of the following transfer equations received TLE-3. Its time complexity is $O(n^2)$.

$$time[i] = \max\{time[j] + \sqrt{\frac{2(y_j - y_i)}{g}}\} + c_i, \quad p_j \to p_i,$$

$$Answer = \max\{time[i] + \sqrt{\frac{2y_i}{g}}\},$$

How to optimize our program? Notice that if Kirill can reach p_j from p_l, and reach p_i from p_j, then p_l won't be the decision point at p_i since passing through p_j must yields a longer duration of falling. Also note that

1 Unsymmetric eigenvaluve problems

Theorem 5 (Gershgorin Circle Theorem)**.** 1) If $X^{-1}AX = D + F$, where $D = \text{diag}(d_1, ...d_n)$ and F has zero diagonal entries, then

$$\sigma(A) \subset \bigcup_{i=1}^{n} D_i, \quad D_i = \{z \in \mathbb{C}, |z - d_i| \le \sum_{j=1}^{n} |f_{ij}|\},$$

2) If the Gershgorin disk D_i is isolated from other disks, then it contains precisely one eigenvalue of A.

Proof. 1) Suppose that $\lambda \in \sigma(A), \lambda \neq d_i, 1 \leq i \leq n$, $D - \lambda I + F = X^{-1}AX - \lambda I$ is singular.

$$(D-\lambda I)^{-1} = \text{diag}(\frac{1}{d_i - \lambda}), \quad G = (D-\lambda I)^{-1}F = (\frac{f_{ij}}{d_i - \lambda})_{1\leq i,j\leq n},$$

We are interested in the l_∞ norm of operator G, which is defined as $\|G\|_\infty = \max_{x\neq 0} \frac{\|Gx\|_\infty}{\|x\|_\infty}$. Suppose $\|x\|_\infty = 1$,

$$Gx = (\sum_j g_{ij}x_j)_{1\leq i\leq n}, \quad \|Gx\|_\infty \leq \max_i \sum_j |g_{ij}| = \max_i \sum_j \frac{|f_{ij}|}{|d_i - \lambda|},$$

and the equality holds. On the other hand, if $A+B$ is singular, A is non-singular, suppose $(A+B)u = 0, u \neq 0$, then

$$Au = -Bu, \quad u = -A^{-1}Bu, \quad \|A^{-1}Bu\|_p = \|u\|_p, \quad \|A^{-1}B\|_p \geq 1,$$

Let $A = D - \lambda I, B = F, p = \infty$, we have $1 \leq \sum_j \frac{|f_{kj}|}{|d_k - \lambda|}$ for some $1 \leq k \leq n$,

$$|d_k - \lambda| \leq \sum_j |f_{kj}|, \quad \lambda \in D_k,$$

Another proof: assume $\lambda \in \sigma(A)$ with eigenvector $u \in \mathbb{C}^n$, $k = \arg\max_j |u_j|$, then

$$|(d_k - \lambda)u_k| = |-\sum_j f_{kj}u_j| \leq |u_k| \sum_j |f_{kj}|, \quad |d_k - \lambda| \leq \sum_j |f_{kj}| = r_k,$$

2) If $D_k \cap D_i = \emptyset, i \neq k$, we show that there exist precisely one $\lambda \in \sigma(A), \lambda \in D_k$ and λ has multiplicity 1. i) Uniqueness: if $\lambda \in D_k$ is an eigenvalue of $D + F$, $u \in \mathbb{C}^n$ is an eigenvector of $D - \lambda I + F$. If $i = \arg\max_j |u_j|, i \neq k$ then

$$(d_i - \lambda)u_i = -\sum_j f_{ij}u_j, \quad |(d_i - \lambda)u_i| > r_i|u_i| \geq \sum_j |f_{ij}||u_j|,$$

contradiction! So we have $k = \arg\max_j |u_j|, \|u\| = |u_k|$. Without loss of generality, assume that $k = n$, $D'_{n-1}, F'_{n-1} \in \text{Aut}(\mathbb{C}^{n-1})$ are the first $n-1$ rows and columns of D, F. Then for $x \in \mathbb{C}^{n-1}$,

$$((D' - \lambda I')^{-1}F'x)_i = \sum_{j=1}^{n-1} \frac{f_{ij}x_j}{d_i - \lambda}, \quad |((D' - \lambda I')^{-1}F'x)_i| \leq \sum_{j=1}^{n-1} \frac{|f_{ij}||x_j|}{|d_i - \lambda|} < \|x\|_\infty,$$

So $\|(D' - \lambda I')^{-1}F'\|_\infty < 1$, and we have the following expansion

$$(D' - \lambda I' + F')^{-1} = (D' - \lambda I')^{-1}(I' + (D' - \lambda I')^{-1}F')^{-1}$$

$$(I' + (D' - \lambda I')^{-1}F')^{-1} = \sum_{m\geq 0}((\lambda I' - D')^{-1}F')^m,$$

The right hand side above is absolutely convergent, so $D' - \lambda I' + F'$ is invertible. It follows that if $\lambda \in D_k$ is an eigenvalue of $D + F$, then it has multiplicity 1.

ii) Resolvent method:

$$R(\lambda) = (D - \lambda I + F)^{-1} : \mathbb{C}^n \to \mathbb{C}^n,$$

As an operator valued function, $R : \mathbb{C} \to \mathrm{Aut}(\mathbb{C}^n)$ is meromorphic in the following sense: for any $\phi \in (\mathbb{C}^n)^*, v \in \mathbb{C}^n$, $f(\phi, v, \lambda) = \langle \phi, R(\lambda)v \rangle$ is a meromorphic function of λ.

$$\langle \phi, R(\lambda_1)v \rangle - \langle \phi, R(\lambda_0)v \rangle = \langle \phi, (R(\lambda_1) - R(\lambda_0))v \rangle = \langle \phi, (\lambda_1 - \lambda_0)R(\lambda_1)R(\lambda_0)v \rangle,$$

$R(\lambda_1), R(\lambda_0)$ are commutative: $R(\lambda_1)R(\lambda_0) = R(\lambda_0)R(\lambda_1)$.

$$\frac{\langle \phi, R(\lambda_1)v \rangle - \langle \phi, R(\lambda_0)v \rangle}{\lambda_1 - \lambda_0} = \langle \phi, R(\lambda_1)R(\lambda_0)v \rangle$$

hence $f(\phi, v, \lambda)$ is meromorphic with $\frac{df(\phi,v,\lambda)}{d\lambda} = \langle \phi, R(\lambda)^2 v \rangle$.

$$\lambda \text{ singular} \iff \|R(\lambda)\| = \infty, \quad \|R(\lambda)\| = \max_{\|\phi\|=\|v\|=1} |\langle \phi, R(\lambda)v \rangle|,$$

In finite dimensional case, $D - \lambda I + F : \mathbb{C}^n \to \mathbb{C}^n$ induces an automorphism on $\bigwedge^n(\mathbb{C}^n)$:

$$e_1 \wedge e_2 ... \wedge e_n \mapsto \bigwedge_{i=1}^{n} (D - \lambda I + F)e_i = \det(D - \lambda I + F)e_1 \wedge e_2 ... \wedge e_n,$$

iii) Consider the eigenvalues of $D + \epsilon F, 0 \le \epsilon \le 1$, by 1), all of its eigenvalues lie in $\bigcup_{i=1}^n \epsilon D_i$. These eigenvalues vary continuously with respect to ϵ, and we may denote them as $\lambda_i(\epsilon), 1 \le i \le n$. When $\epsilon = 0$, we have $\lambda_i(0) = d_i$, so by continuity argument and the fact that $D_k \cap D_i = \emptyset, i \neq k$, we know that there is precisely one eigenvalue $\lambda_k(\epsilon) \in \epsilon D_k$. Take $\epsilon = 1$ finishes the proof. □

Theorem 6 (Bauer-Fike)**.** If μ is an eigenvalue of $A+E \in \mathbb{C}^{n*n}$ and $X^{-1}AX = D = \mathrm{diag}(\lambda_q, ..., \lambda_n)$, then

$$\min_{\lambda \in \sigma(A)} |\lambda - \mu| \le \kappa_p(X)\|E\|_p,$$

Proof. It suffices to consider the case $\mu \notin \sigma(A)$. If the matrix $X^{-1}(A + E - \mu I)X$ is singular, then so is $I + (D - \mu I)^{-1}X^{-1}EX$. Then we have

$$1 \le \|(D - \mu I)^{-1}X^{-1}EX\|_p \le \|(D - \mu I)^{-1}\|_p \|X\|_p \|E\|_p \|X^{-1}\|_p,$$

Since$\|(D - \mu I)^{-1}\|_p = \max_{\lambda \in \sigma(A)} \frac{1}{|\lambda - \mu|}$, we have finished our proof. □

Definition 2. For square matrix A define the condition number $\kappa(A) = \|A\|\|A^{-1}\|$, with the convention that $\kappa(A) = \infty$ for singular A. $\kappa(\cdot)$ depends on the underlying norm and subscripts are used accordingly.

$$\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}, \quad \frac{1}{\kappa_p(A)} = \min_{A+\Delta A \text{ singular}} \frac{\|\Delta A\|_p}{\|A\|_p},$$

$$\kappa(A) = \lim_{\epsilon \to 0} \sup_{\|\Delta A\| \le \epsilon \|A\|} \frac{\|(A + \Delta A)^{-1} - A^{-1}\|}{\epsilon \|A^{-1}\|}$$

Theorem 7. Let $Q^H AQ = D + N$ be a Schur decomposition of $A \in \mathbb{C}^{n*n}$, i.e., $Q \in \mathbb{C}^{n*n}$ is unitary, $D = \mathrm{diag}(\lambda_1, ..., \lambda_n)$ and $N \in \mathbb{C}^{n*n}$ is strictly upper diagonal. Q can be chosen so that the eigenvalues λ_i appear in any order along the diagonal. If $\mu \in \sigma(A + E)$ and p is the smallest positive integer such that $N^p = 0$, then

$$\min_{\lambda \in \sigma(A)} |\lambda - \mu| \le \max\{\theta, \theta^{\frac{1}{p}}\}, \quad \theta = \|E\|_2 \sum_{k=0}^{p-1} \|N\|_2^k,$$

Extreme eigenvalue sensitivity for a matrix A cannot occur if A is normal. But A nonnormal matrix can have a mixture of well-conditioned and ill-conditioned eigenvalues. Suppose that λ is a simple eigenvalue of $A \in \mathbb{C}^{n*n}$ and that x and y satisfy $Ax = \lambda x, y^H A = \lambda y^H, \|x\|_2 = \|y\|_2 = 1$. If $Y^H AX = J$ is the Jordan decomposition with $Y^H = X^{-1}$, then y and x are nonzero multiples of $X(:, i), Y(:, i)$ for some i, so $y^H x \neq 0$.

$$(A + \epsilon F)x(\epsilon) = \lambda(\epsilon)x(\epsilon), \quad \|F\|_2 = 1,$$

We refer to the reciprocal of $s(\lambda) = |y^H x|$ as the condition of the eigenvalue λ. A small $s(\lambda)$ implies that A is near a matrix having a multiple eigenvalue. In particular, if λ is distinct and $s(\lambda) < 1$, then there exists an E such that λ is a repeated eigenvalue of $A + E$ and

$$\frac{\|E\|_2}{\|A\|_2} \leq \frac{s(\lambda)}{\sqrt{1 - s(\lambda)^2}},$$

In general, if λ is a defective eigenvalue of A, then $O(\epsilon)$ perturbations in A can result in $O(\epsilon^{\frac{1}{p}})$ perturbations in λ if λ is associated with a p-dimensional Jordan block.

2 Symmetric eigenvalue problems

Theorem 8 (Gershgorin)**.** A is real symmetric, Q is orthogonal, if $Q^t AQ = D+F, D = \mathrm{diag}(d_1, d_2, ..., d_n)$ and F has zero diagonal entries, then

$$\sigma(A) \subset \bigcup_{i=1}^{n} [d_i - r_i, d_i + r_i], \quad r_i = \sum_j |f_{ij}|,$$

Proof. Exactly the same as the unsymmetric case, with an additional property that $\sigma(A) \subset \mathbb{R}$. □

Theorem 9 (Wielandt-Hoffman)**.** If A and $A + E$ are $n * n$ symmetric matrices, then

$$\sum_{i=1}^{n} (\lambda_i(A + E) - \lambda_i(A))^2 \leq \|E\|_F^2 = \sum_{i,j=1}^{n} |e_{ij}|^2$$

Theorem 10. If A and $A + E$ are $n * n$ symmetric matrices, then

$$\lambda_k(A) + \lambda_n(E) \leq \lambda_k(A + E) \leq \lambda_k(A) + \lambda_1(E), \quad 1 \leq k \leq n,$$

$$|\lambda_k(A + E) - \lambda_k(A)| \leq \|E\|_2 = \max\{|\lambda_n(E)|, |\lambda_1(E)|\}, \quad 1 \leq k \leq n,$$

Theorem 11 (Interlacing property)**.** If $A \in \mathbb{R}^{n*n}$ is symmetric and $A_r = A(1 : r, 1 : r)$, then

$$\lambda_{r+1}(A_{r+1}) \leq \lambda_r(A_r) \leq \lambda_r(A_{r+1}) \leq ... \leq \lambda_2(A_{r+1}) \leq \lambda_1(A_r) \leq \lambda_1(A_{r+1}), \quad 1 \leq r \leq n - 1,$$

Theorem 12. Suppose $B = A + \tau cc^t, A \in \mathbb{R}^{n*n}, A = A^t, c \in \mathbb{R}^n, \|c\|_2 = 1, \tau \in \mathbb{R}$. we have

$$\lambda_i(B) \in [\lambda_i(A), \lambda_{i-1}(A)], 2 \leq i \leq n, \text{ when } \tau \geq 0,$$

$$\lambda_i(B) \in [\lambda_{i+1}(A), \lambda_i(A)], 1 \leq i \leq n - 1, \text{ when } \tau < 0,$$

In either case, there exist $m_1, m_2, ..., m_n \geq 0, m_1 + m_2 + ... + m_n = 1$ such that

$$\lambda_i(B) = \lambda_i(A) + m_i\tau, \quad 1 \leq i \leq n,$$

Proposition 1. 1) If $T = QR$ is the QR factorization of a symmetric tridiagonal matrix $T \in \mathbb{R}^{n*n}$, then Q has lower bandwidth 1 and R has upper bandwidth 2 and it follows that $T_+ = RQ = Q^tTQ$ is also symmetric and tridiagonal.

2) If $s \in \mathbb{R}$ and $T - sI = QR$ is the QR factorization, then $T_+ = RQ + sI = Q^tTQ$ is also tridiagonal. This is called a shifted QR step.

3) If T is unreduced, then the first $n-1$ columns of $T - sI$ are independent regardless of s.

4) If $T \in \mathbb{R}^{n*n}$ is tridiagonal, then its QR factorization can be computed by applying a sequence of $n-1$ Givens rotations.

3 Solve univariate polynomial equations using $SL_2(\mathbb{R})$

Assume that a degree 4 real coefficient polynomial $P(x) = x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ has no real roots. Assume that its four roots are

$$x_1, \overline{x_1}, x_2, \overline{x_2}, \quad x_1, x_2 \in \mathbb{H}, \quad \Im(x_1) > 0, \Im(x_2) > 0,$$

Assume that $x_1 = u_1 + v_1 i, x_2 = u_2 + v_2 i$. Half circle arc on the upper half plane which passes x_1, x_2 and meets real axis orthogonally is uniquely determined. Assume its center is $(u, 0)$, then

$$(u - u_1)^2 + v_1^2 = (u - u_2)^2 + v_2^2, \quad u = \frac{u_2^2 + v_2^2 - u_1^2 - v_1^2}{2(u_2 - u_1)},$$

Radius of the half circle arc is

$$\begin{aligned} r_0^2 &= (u - u_1) + v_1^2 = \frac{(u_2^2 - 2u_1u_2 + u_1^2 + v_2^2 - v_1^2)^2}{4(u_2 - u_1)^2} + v_1^2 \\ &= \frac{((u_2 - u_1)^2 + v_2^2 - v_1^2)^2 + 4(u_2 - u_1)^2 v_1^2}{4(u_2 - u_1)^2} = \frac{(u_2 - u_1)^4 + 2(u_2 - u_1)^2(v_2^2 + v_1^2) + (v_2^2 - v_1^2)^2}{4(u_2 - u_1)^2}, \end{aligned}$$

The action $SL_2(\mathbb{R}) \curvearrowright \overline{\mathbb{C}}$ is given by

$$x \mapsto g(x) = \frac{px + q}{rx + s}, \quad g = \begin{pmatrix} p & q \\ r & s \end{pmatrix} \in SL_2(\mathbb{R}),$$

Its kernel is $\pm id$. We want to determine g such that

$$\frac{px_1 + q}{rx_1 + s} = i, \quad \frac{px_2 + q}{rx_2 + s} \in \mathbb{R}_+ i,$$

4 Sendov's Conjecture

Conjecture 1 (Sendov's Conjecture)**.** For a polynomial $f(z) = (z - r_1)(z - r_2)...(z - r_n), n \geq 2$ with all roots $r_1, r_2, ..., r_n$ inside the closed unit disk $\{|z| \leq 1\}$, each of the n roots is at a distance no more than 1 from at least one root of $f'(z)$.

It suffices to show that for a fixed r_1, the following distance function has maximum no more than 1.

$$d(r_2, ..., r_n) = \min |r_1 - \xi_i|, \quad f'(z) = (z - \xi_1)(z - \xi_2)...(z - \xi_{n-1}),$$

Two near counter-examples are

$$f_1(z) = z^n - 1, \quad r_1 = e^{\frac{2\pi i}{n}}, \quad f_2(z) = z^n - z, \quad r_1 = 0,$$

In the latter case the distance from 0 to any root of $f_2'(z)$ is $n^{-\frac{1}{n-1}} = 1 - O(\frac{\log n}{n})$,