How to Think Mathematically?

- Ayushi Mittal

Learn foundational Mathematics topics

- Read books and / or blogs on topics that strengthen core mathematical concepts like linear algebra, geometry, graph theory, number theory, combinatorics, probability etc.
- Do not black box basic topics. Understand how certain algorithms work, why they are correct and how they achieve their time complexity.
- After reading theory, practice a topic thoroughly by its tag with increasing difficulty.

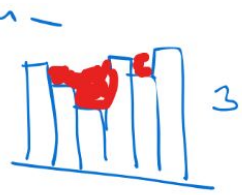
Practice identifying patterns

- While solving a mathematics based problem, it is a good idea to search for patterns when stuck on a dead end.
- Practise writing brute force code to generate solutions for small test cases and find patterns within them.
- Once you find a pattern you now have a possible solution instead of a dead end and you can think on proving or disproving why it works.

Problem 1

ICPC Amritapuri Regionals 2025: Expected Rain

https://drive.google.com/drive/folders/1XiOQybfPXgWI_2CzQ1O74rrvwpW8GqtL



1 2 3
1
3 | — 6
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

P(x) →

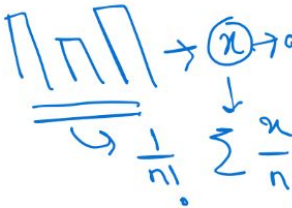
E(x) = x · P(x)

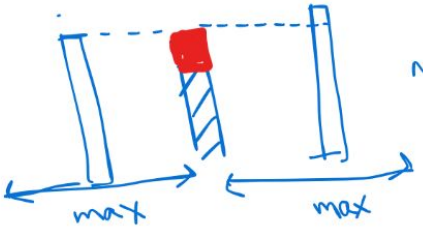
Dice : 1 2 3 4 5 6

P(2) = 1/6

E(2) = 2 × 1/6 = 1/3

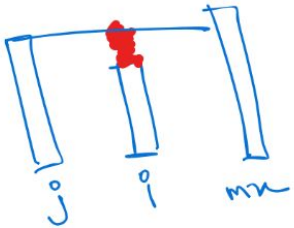
E = 1·1/6 +
2·1/6 +
3·1/6 ...





min(left max, right max) − h_i =

either left max or right max is the maximum of the whole array



(i < j)

j ≤ mx

Total rain water for all permutations

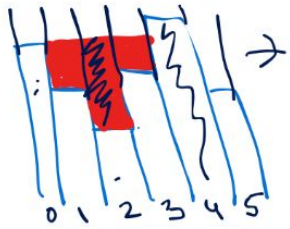
↓ no of permutations → n!

take a pair a_i, a_j such that

$a_i \le a_j \le max$

↳ $(a_j - a_i)$ = rainwater

find all permutation where a_j is maxleft / maxright for a_i



→ calculate how much water is above each column.

l	X	0	0	0	X	4	$\min(h_l, h_r) - h_i$
r	4	4	4	4	X	X	

for each i, fix j

how many permutations are there s.t.

q_j is leftmax

$q_i^0 = P(j, i)$

compute this using brute force

$$\sum_{j=1}^{n-1} \sum_{i=0}^{j-1} (q_j - a_{ij}) \cdot P(j, i) \quad (q_j \geq q_i)$$

$$\left(\frac{1}{n-j+1} \cdot \frac{1}{n-j-1} \right)$$

Problem 2 solution

```
void Solve() {
    int n; cin >> n;
    vector <int> h(n);
    for (auto &x : h) cin >> x;

    int ans = 0;

    sort(h.begin(), h.end());
    int mx = h[n - 1];
    int sum = accumulate(h.begin(), h.end(), 0LL);
    sum %= mod;

    for (int i = 0; i < n; i++){
        int prob = inv(n - i);
        int ex = 1 + i * inv(n - i + 1) % mod;
        ans += ex * prob % mod * h[i] % mod;
    }

    ans %= mod;
    ans *= 2;

    ans -= sum;
    ans -= mx * n;
    ans %= mod;
    if (ans < 0) ans += mod;

    cout << ans << "\n";
}
```

Prove your solution

- Prove your solution during or after the contest and always while practising.
- Proving your solutions while you are not deeply familiar with a topic helps develop your intuition.
- Better intuitions means in the future you can think towards the right solution faster and detect patterns quicker.

Problem 2

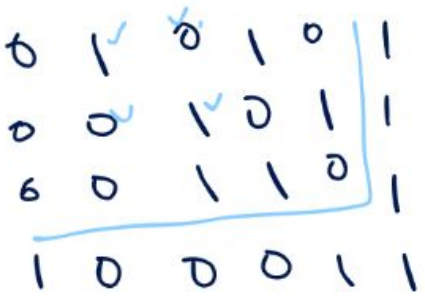
ICPC Amritapuri Regionals 2025: Flip on Cycle

https://drive.google.com/drive/folders/1XiOQybfPXgWI_2CzQ1O74rrvwpW8GqtL

0 → don't flip

1 → need to flip

A	B	→	C
0	0		0
1	0		1



✓

X

X

last column which is not in last row

we can do something similar for last row which is not last column

→ element at last row, last column will always = 0 or answer doesn't exist

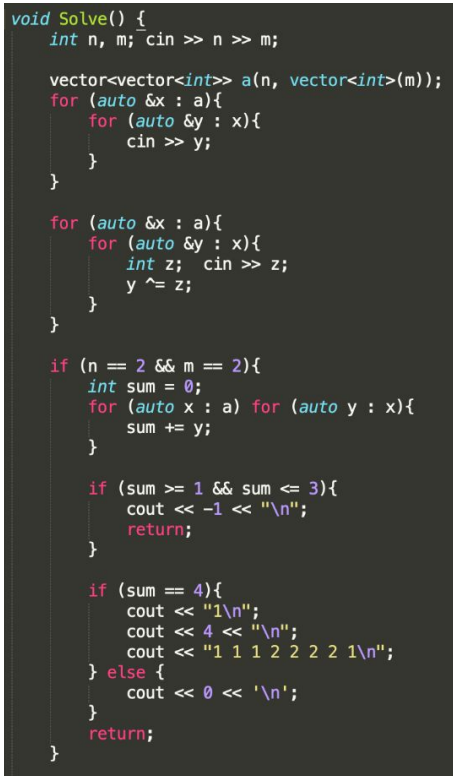
1 0 0 0

0 0 0 0 → all 0s!

0 0 0 0

odd flips → answer never exists

Problem 3 solution



```cpp
void Solve() {
    int n, m; cin >> n >> m;

    vector<vector<int>> a(n, vector<int>(m));
    for (auto &x : a){
        for (auto &y : x){
            cin >> y;
        }
    }

    for (auto &x : a){
        for (auto &y : x){
            int z;  cin >> z;
            y ^= z;
        }
    }

    if (n == 2 && m == 2){
        int sum = 0;
        for (auto x : a) for (auto y : x){
            sum += y;
        }

        if (sum >= 1 && sum <= 3){
            cout << -1 << "\n";
            return;
        }

        if (sum == 4){
            cout << "1\n";
            cout << 4 << "\n";
            cout << "1 1 1 2 2 2 2 1\n";
        } else {
            cout << 0 << '\n';
        }
        return;
    }
```

Problem 3 solution

```
vector<vector<int>> ans;

auto work = [&](vector <int> v){
    ans.push_back(v);

    int k = (int)v.size() / 2;
    for (int i = 0; i < k; i++){
        int x = v[2 * i];
        int y = v[2 * i + 1];

        a[x][y] ^= 1;
    }
};

if (n == 2){
    for (int i = 0; i < m; i++){
        if (a[0][i] != a[1][i]){
            cout << -1 << "\n";
            return;
        }
    }

    for (int i = 0; i < m; i++){
        if (a[0][i] == 1){
            if (i + 1 == m){
                work({0, i, 0, i - 1, 0, i - 2, 1, i - 2, 1, i - 1, 1, i});
                work({0, i - 1, 0, i - 2, 1, i - 2, 1, i - 1});
            } else {
                work({0, i, 0, i + 1, 1, i + 1, 1, i});
            }
        }
    }
} else if (m == 2){
```

Problem 3

```
} else if (m == 2){
    for (int i = 0; i < n; i++){
        if (a[i][0] != a[i][1]){
            cout << -1 << "\n";
            return;
        }
    }

    for (int i = 0; i < n; i++){
        if (a[i][0] == 1){
            if (i + 1 == n){
                work({i, 0, i - 1, 0, i - 2, 0, i - 2, 1, i - 1, 1, i, 1});
                work({i - 1, 0, i - 2, 0, i - 2, 1, i - 1, 1});
            } else {
                work({i, 0, i + 1, 0, i + 1, 1, i, 1});
            }
        }
    }
} else {
    int sum = 0;
    for (int i = 0; i < n; i++){
        for (int j = 0; j < m; j++){
            sum += a[i][j];
        }
    }

    if (sum % 2 == 1){
        cout << -1 << "\n";
        return;
    }
```

Problem 3

```
    for (int i = 0; i < n; i++){
        for (int j = 0; j < m; j++){
            if (a[i][j] == 1){
                if (j + 1 != m){
                    // swap it with next
                    if (i + 1 != n){
                        work({i, j, i, j + 1, i + 1, j + 1, i + 1, j});
                    } else {
                        work({i, j, i, j + 1, i - 1, j + 1, i - 2, j + 1, i - 2, j, i - 1
                        work({i - 1, j + 1, i - 2, j + 1, i - 2, j, i - 1, j});
                    }
                } else {
                    work({i, j - 2, i, j - 1, i, j, i + 1, j, i + 1, j - 1, i + 1, j - 2}
                    work({i, j - 2, i, j - 1, i + 1, j - 1, i + 1, j - 2});
                }
            }
        }
    }
}

cout << ans.size() << "\n";

for (auto x : ans){
    int sz = x.size();
    cout << (sz / 2) << "\n";
    for (auto y : x){
        cout << (y + 1) << " ";
    }
    cout << "\n";
}
```

Develop a problem solving framework

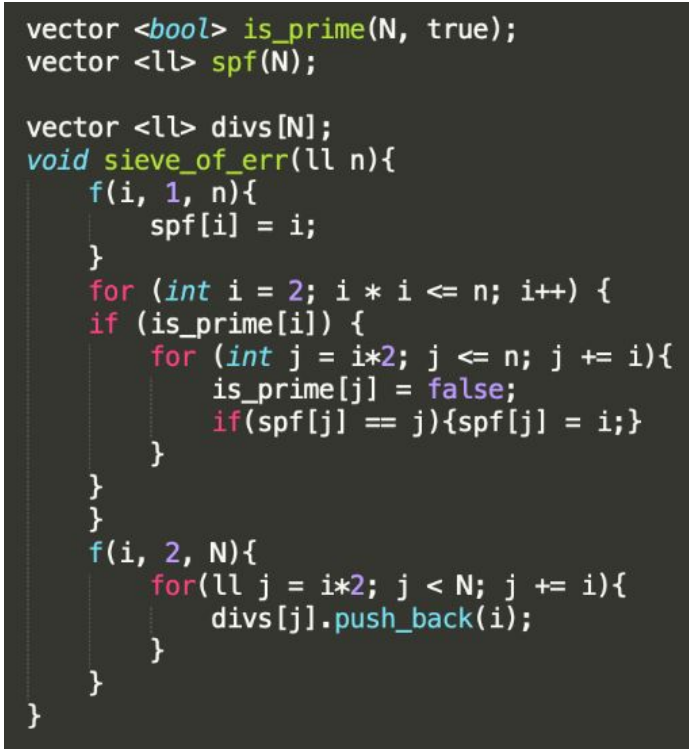
- Understand the problem and carefully analyse the sample cases
- Break down the problem to further subproblems if you can
- For a particular subproblem, reframe it into something simpler
- Write down all the mathematical relations you can find and narrow down your solution with your constraints
- Tie up the solutions of all your subproblems together

Problem 3

Refact.ai Match 1 (Codeforces Round 985): Common Generator

https://codeforces.com/contest/2029/problem/E

Problem 4 Solution



```
vector <bool> is_prime(N, true);
vector <ll> spf(N);

vector <ll> divs[N];
void sieve_of_err(ll n){
    f(i, 1, n){
        spf[i] = i;
    }
    for (int i = 2; i * i <= n; i++) {
    if (is_prime[i]) {
        for (int j = i*2; j <= n; j += i){
            is_prime[j] = false;
            if(spf[j] == j){spf[j] = i;}
        }
    }
    }
    f(i, 2, N){
        for(ll j = i*2; j < N; j += i){
            divs[j].push_back(i);
        }
    }
}
```

Problem 4 Solution

```
void solve(ll tt, ll TT){
    ll n; cin>>n;
    vector <ll> a(n);
    f(i, 0, n){cin>>a[i];}
    ll p = 0;
    f(i, 0, n){
        if(is_prime[a[i]]){
            if(p && a[i] != p){cout<<-1<<"\n"; return;}
            p = a[i];
        }
    }
    if(!p){cout<<2<<"\n"; return;}
    f(i, 0, n){
        if(a[i] < p){cout<<-1<<"\n"; return;}
        if(a[i] % p == 0){continue;}
        ll s = spf[a[i]];
        if(a[i] / s >= p){continue;}
        a[i] -= s;
        s = spf[a[i]];
        if(a[i] / s >= p){continue;}
        cout<<-1<<"\n"; return;
    }
  cout<<p; br;
  return;
}
```