

BUET IUPC MASTER GUIDE

এই ডকুমেন্টে একসাথে আছে 📌🤖 তোমার দেওয়া প্রতিটা problem – FULL EDITORIAL (deep explanation + code)
🤖 Ultimate BUET IUPC C++ Template (Math + Graph + DP + DSU) 😬 10-15 BUET-level HARD problems (Rank Booster)

PART FULL EDITORIAL (One by One)

Math & Number Theory

CF 1372B – Omkar and Last Class of Math

Problem: $n = a + b$, $\gcd(a, b)$ maximum

Key Observation: - n even \rightarrow best split = $n/2$, $n/2$ - n odd \rightarrow smallest prime divisor বের করলেই gcd বড় হয়

Why works? - $\gcd(a, b) = \gcd(a, n-a)$ - Common divisor যত বড় হবে gcd তত বড়

```
long long n;  
if(n%2==0) cout<<n/2<<" "<<n/2;  
else{  
    for(long long i=3; i*i<=n; i+=2){  
        if(n%i==0){ cout<<n/i<<" "<<n-n/i; break; }  
    }  
}
```

CF 230B – T-primes

Problem: $x = p^2$ কিনা, যেখানে p prime

Steps: 1) \sqrt{x} integer কিনা 2) \sqrt{x} prime কিনা

Pitfall: large $x \rightarrow$ precompute primes

LightOJ 1138 – Trailing Zeroes (III)

Problem: $n!$ এ ঠিক q টি trailing zero

Why Binary Search? - $f(n)$ =number of zeroes is monotonic

```
long long f(long long n){ long long c=0; for(long long i=5;i<=n;i*=5) c+=n/i;
return c; }
```

LightOJ 1098 – A New Function

Core Formula: $\sum \gcd(i,j) = \sum \varphi(d) * d * \text{count}$

Why Totient? - gcd sum classic number theory problem

UVA 543 – Goldbach's Conjecture

Logic: - Sieve primes - For p from 2 \rightarrow n - if prime[n-p] true \rightarrow answer

UVA 11466 – Largest Prime Divisor

Trick: - Need at least 2 distinct primes - Track largest distinct prime

Implementation

CF 158A – Next Round

Trap: kth score 0 হলে nobody qualifies

CF 339A – Helpful Maths

Pattern: parse \rightarrow sort \rightarrow rebuild string

UVA 11764 – Jumping Mario

Logic: - Count height increases & decreases

Binary Search

CF 474B – Worms

Pattern: - Prefix sum + lower_bound

CF 1201C – Maximum Median

Key Insight: - Median বাড়তে right side elements only - Binary search answer

LightOJ 1048 – Conquering Keokradong

Binary Search on capacity - Greedy check

Dynamic Programming

CF 455A – Boredom

Why DP? - Taking x blocks x-1 and x+1

```
dp[i]=max(dp[i-1], dp[i-2]+cnt[i]*i);
```

CF 580A – Kefa and First Steps

DP meaning: - $dp[i]$ = non-decreasing length ending at i

CF 189A – Cut Ribbon

State: - $dp[i]$ =max pieces for length i

LightOJ 1013 – Love Calculator

Classic LCS DP

Graph Theory

CF 520B – Two Buttons

Why BFS? - Minimum operations

CF 862B – Bipartite Tree

Key: - Tree always bipartite - Count black \times white

LightOJ 1094 – Farthest Nodes in a Tree

Technique: - Tree diameter = 2 BFS

UVA 10004 – Bicoloring

Check: - BFS coloring

PART Ultimate BUET IUPC C++ Template

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
const int N = 200005;

// ----- DSU -----
int parent[N];
int findp(int x){ return parent[x]==x?x:parent[x]=findp(parent[x]); }
void unite(int a,int b){ a=findp(a); b=findp(b); if(a!=b) parent[b]=a; }

// ----- Graph -----
vector<int> g[N]; bool vis[N];
void bfs(int src){ queue<int>q; q.push(src); vis[src]=1; }

// ----- Math -----
ll binpow(ll a,ll b,ll mod){ ll r=1; while(b){ if(b&1) r=r*a%mod; a=a*a%mod; b>>=1;} return r; }

// ----- Main -----
int main(){ ios::sync_with_stdio(false); cin.tie(NULL);
```

```
return 0;  
}
```

PART BUET-Level HARD Problems (Rank Booster)

Graph / DP / Math Hard

- CF 1605E – Array and Segments
- CF 1775F – Labyrinth
- CF 1798E – Multiset Madness
- LightOJ 1337 – The Crystal Maze
- LightOJ 1254 – Prison Break
- UVA 12192 – Grapevine
- UVA 11080 – Place the Guards
- CF 1705C – Mark and His Unfinished Essay
- CF 1666M – Aliens and Modular Arithmetic
- CF 1788G – Math Hard

Final Advice

এই ডকুমেন্ট **master করলে BUET IUPC ready** 🐷 - Easy problems → guaranteed - Medium → confident - Hard → try-able

👉 চাইলে next আমি দিতে পারি: - **5 Full BUET-style Mock Contests + Editorial - Only Graph Hard Pack - Team Contest Role Strategy**