BUET IUPC Complete Notes

এই নোটগুলো **BUET IUPC** (Inter University Programming Contest) এর জন্য ফুল সিলেবাস কভার করে বানানো। প্রতিটা টপিকের সাথে **concept + tricks + sample code (C++17)** দেওয়া আছে।

---

1. Time & Space Complexity

Concepts

- Big-O, Big-Theta, Big-Omega
- Worst / Average / Best case
- Constraints অনুযায়ী algorithm নির্বাচন

Rules of Thumb

- n ≤ 1e5 → O(n log n)
- n ≤ 1e6 → O(n)
- n ≤ 1e7 → O(n)
- n ≤ 500 → O(n^3)

---

2. STL (C++ Standard Template Library)

Important Containers

- vector, deque, list
- stack, queue, priority_queue
- set, multiset, map, unordered_map

```
vector<int> v = {1,2,3};
sort(v.begin(), v.end());
map<int,int> mp;
mp[5]++;
```

Useful Algorithms

```
sort(v.begin(), v.end());
binary_search(v.begin(), v.end(), x);
__gcd(a,b);
```

---

3. Mathematics

Number Theory

- GCD, LCM
- Prime sieve
- Modular arithmetic
- Fast exponentiation

```
long long binpow(long long a, long long b, long long mod){
    long long res = 1;
    while(b){
        if(b & 1) res = res * a % mod;
        a = a * a % mod;
        b >>= 1;
    }
    return res;
}
```

Sieve of Eratosthenes

```
const int N = 1e6;
vector<bool> isPrime(N,true);
void sieve(){
    isPrime[0]=isPrime[1]=false;
    for(int i=2;i*i<N;i++)
        if(isPrime[i])
            for(int j=i*i;j<N;j+=i)
                isPrime[j]=false;
}
```

---

4. Recursion & Backtracking

Examples

- Subset generation
- Permutations
- N-Queen

```
void gen(int i){
    if(i==n){
        // process
        return;
```

```
    }
    gen(i+1);
}
```

---

5. Sorting & Searching

Sorting

- Bubble, Selection, Insertion (basic)
- Merge Sort
- Quick Sort

Binary Search

```
int l=0,r=n-1;
while(l<=r){
    int mid=(l+r)/2;
    if(a[mid]==x) break;
    else if(a[mid]<x) l=mid+1;
    else r=mid-1;
}
```

---

6. Greedy Algorithms

Common Problems

- Activity selection
- Interval scheduling
- Coin change (when applicable)

Pattern

- Sort by criteria
- Pick best local option

---

7. Dynamic Programming (DP)

Types

- 1D DP
- 2D DP

- Knapsack
- LIS

```
vector<int> dp(n,1);
for(int i=0;i<n;i++)
  for(int j=0;j<i;j++)
    if(a[j]<a[i])
      dp[i]=max(dp[i],dp[j]+1);
```

---

8. Graph Theory

Representation

```
vector<int> g[N];
```

BFS

```
queue<int> q;
q.push(src);
while(!q.empty()){
  int u=q.front(); q.pop();
  for(int v:g[u]){
    if(!vis[v]){
      vis[v]=1;
      q.push(v);
    }
  }
}
```

DFS

```
void dfs(int u){
  vis[u]=1;
  for(int v:g[u])
    if(!vis[v]) dfs(v);
}
```

Shortest Path

- Dijkstra
- Floyd Warshall

9. Disjoint Set Union (DSU)

```
int parent[N];
int find(int x){
  if(parent[x]==x) return x;
  return parent[x]=find(parent[x]);
}
void unite(int a,int b){
  a=find(a); b=find(b);
  if(a!=b) parent[b]=a;
}
```

---

10. Strings

Topics

- Frequency counting
- Palindrome
- Pattern matching
- KMP (basic)

```
string s;
reverse(s.begin(), s.end());
```

---

11. Bit Manipulation

```
if(x & (1<<i)){}
x |= (1<<i);
x ^= (1<<i);
```

---

12. Geometry (Basic)

- Distance between points
- Orientation
- Area

```cpp
double dist(double x1,double y1,double x2,double y2){
  return sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
}
```

---

13. Contest Tips (BUET IUPC)

- Easy → Medium → Hard order
- Avoid overthinking
- Read constraints carefully
- Use fast I/O

```cpp
ios::sync_with_stdio(false);
cin.tie(NULL);
```

Final Advice

এই নোটগুলো **2–3 মাস practice** করলে BUET IUPC এর জন্য strong foundation তৈরি হবে। প্রতিটা টপিকের সাথে **online judge (Codeforces, LightOJ, UVA)** থেকে problem solve করো।

Want next: - BUET IUPC previous year problem analysis - Topic-wise practice problem list - 30/60 days contest roadmap