Chittagong University of Engineering & Technology
Dept. of Computer Science & Engineering

CUET Competitive Programmer's Syllabus

A guideline for the students of CUET for
'Competitive Programming'

CUET Computer Club

Prepared By

Md. Billal Hossain (Lecturer, Dept. of CSE, CUET) Omar Sharif (Lecturer, Dept. of CSE, CUET)
Jannatul Ferdows Jenny ('12), Tahsin Rahman ('13), Yamin Iqbal ('14)
Rian ('15), Avijeet ('15), Ishtiaq ('16), Mehedi ('16), Eftikhar ('16), Rony ('16), Tapojit ('16), Rajon ('16), Tajir ('16), Maruf ('16), Sayem ('16)
Ahasan ('17), Fariha ('17), Robin ('17), Shawly ('17), Tasfia ('17)

CUET Competitive Programmer's Syllabus

About:

This is a guideline for the Competitive Programmers of CUET in accordance with their academic calendar. It contains the topics that must be covered within first two years if you want to qualify for the world finals. The syllabus is divided into four terms. Each term covers some topic containing detailed tutorials and problem links. Level-1 Term-1 covers the basics of programming Language. Level-1 Term-2 is the most crucial part since it introduces the basic idea of data structures/algorithms. In Level-2, intermediate/some advanced topics are covered. The syllabus of Level-3 and Level-4 is not provided here. It's your own duty to find out your lacking and focus/specialize on them. The links provided here will help you in a great deal but there might be many more resources. So always keep yourself up-to-date.

Goal:

To see CUETians in ICPC World Finals. ☺

Shortcuts:

Some Important Links:

1. Code Templates
2. Resources in Bengali
3. Books Related to Competitive Programming

Contact:

For any constructive feedback send an email to

1. Md. Billal Hossain (mhbillal@cuet.ac.bd)
2. Omar Sharif (omar.sharif@cuet.ac.bd)

Term wise Syllabus

Level-1 Term-1

Tentative Class Schedule:

Week	Topic
1	Variables, Data Types, Scanf/Printf, Format Specifier
2	If-else, Nested If-else
3	Loop Basics: While, Do-while, For-loop
4	Nested Loop, Break, Continue
5	Loop Advanced: Loop+If-else
Online Contest-1	
6	Array: 1D, 2D
7	Character Array, String
8	Function, Recursion
9	Binary Search + Others
10	STL Basic
Online Contest-2	

Guidelines:

- Get complete idea about C
- Get basic idea about C++
- Try to implement what you think
- Must participate in Long-Contests (Upto 5 days) that will be held after each class
- Take help from your Batch-mates, Seniors and Teachers if needed

Expectations:

- Able to implement your idea
- Familiarize with all online judges
- Read Tamim Shahriar Subeen's Book completely
- Solve all the problems of LightOJ beginner's category
- Total Number of problems solved: 200+
- CF rating:
  - -At the start: 900+
  - -At the end: 1200+

Tips:

- Learn about different good resources as well. Geeksforgeeks, emaxx, competitive programming 3, Mahbubul Hasan Shanto vai's book on programming contest, Shafaet Ashraf vai's blog. These are good, but not just these-there are others. Google is your friend.
- Spend around an hour or two at each problem before searching for solutions online. Not more than that, because what is important for you at this stage is to solve more and more problems, and if you spend all your day at one problem, your overall solve count will be low and you will not learn much. But make sure that you spend this time wisely-when we say we have worked for an hour, what happens in reality is that we have worked for just 20 minutes in total, and the other 40 minutes have been wasted away via other 'useful' procrastination. Perhaps learn to adopt the Pomodoro method.
- After seeing the solution to a problem, make sure to understand the solution fully and absolutely. There should be no doubt regarding anything with the solution, and if there is, that means you have not understood it fully/internalized it. Problem solving is more about pattern matching, and being able to match an unseen problem with problems that you have seen before-but if you haven't internalized the solution to the problems that you couldn't solve by yourself, you have then actually just memorized it, and memorized stuff doesn't actually stay in your mind for long-and you won't be able to recognize patterns later as well.

Tutorials & Problems List for Level-1 Term-1

(1-1) Week 1 (Variables, Data types, Scanf/Printf, Format Specifier)

Tutorials:

1. [প্রোগ্রামিং বইঃ অধ্যায় দুই] ডাটা টাইপ, ইনপুট ও আউটপুট।
2. List of all format specifiers in C programming
3. Data Types in C
4. https://www.youtube.com/watch?v=e9Eds2Rc_x

Problems:

1. Codeforces: 1A, 99999100
2. Atcoder: ABC153A, ABC-148A
3. Hackerrank: Playing with Characters
4. URI: 1001,1002,1006,1015,1019

(1-1) Week 2 (If else, Nested If else)

Tutorials:

1. Decision Making in C / C++ (if , if..else, Nested if, if-else-if)
2. কম্পিউটার প্রোগ্রামিং বই: [প্রোগ্রামিং বইঃ অধ্যায় তিন] কন্ডিশনাল লজিক।

Problems:

1. Codeforces: 122A, 263A
2. Codechef: LEAP, EO, ASET002
3. Atcoder: ABC152A, ABC065A
4. URI: 2582
5. UVA: 11172
6. Hackerrank: Conditional Statements in C

(1-1) Week 3 (Loop Basics: While, Do-While, For-loop)

Tutorials:

1. [প্রোগ্রামিং বইঃ অধ্যায় চার] লুপ (Loop)।
2. Loops in C and C++

Problems:

1. DimikOJ: Problem 6, Problem 3, Problem 5
2. Timus: 1149
3. LightOJ: 1000, 1022, 1216
4. Codeforces: 99999123, 791A, 977A

(1-1) Week 4 (Nested Loop, Break, Continue)

Tutorials:

1. Nested Loops in C with Examples
2. Difference between continue and break statements in C++

Problems:

1. CodeChef: BICBPAT1, PPATTERN
2. URI: 1189, 1190, 1183, 1435, 1478, 1186, 1188
3. Hackerrank: Printing Pattern using Loops

(1-1) Week 5 (Loop Advanced: Loop + If-else)

Tutorials:

1. [প্রোগ্রামিং বইঃ অধ্যায় চার] লুপ (Loop) I

Problems:

1. Codechef: LEAPY
2. Atcoder: 079A
3. Timus: 1083, 1209
4. LightOJ: 1001, 1015, 1053, 1008
5. URI: 1557, 1187

(1-1) Week 6 (Array: 1D, 2D)

Tutorials:

1. অ্যারে - Array Archives - হাসানের রাফখাতা
2. Fred: C++ Notes: Table of Contents (See the Arrays section)
3. অ্যারে কম্প্রেশন
4. 1-D Tutorials & Notes | Data Structures (try to solve some problem from this problem section)
5. How to visualize multidimensional arrays
6. Array Data Structure
7. Lecture Notes on Arrays

Problems:

1. UVA: 10038 (Jolly Jumpers), 414 (Machined Surfaces), 665 (False coin), 467 (Synching Signals), 10703 (Free spots), 10855 (Rotated square), 10920 (Spiral Tap),11349 (Symmetric Matrix), 11581 (Grid Successors), 12187 (Brothers)
2. Codeforces: 1213B, 1265B, 1279C

3. Codechef: SALARY

(1-1) Week 7 (Character Array, String)

Tutorials:

1. Strings in C
2. Strings in C (With Examples)
3. C++ Strings
4. string - C++ Reference
5. C++ Strings

Problems:

1. Codeforces: 71A, 1146A, 118A, 1139A, 1223B, 1243B1, 448B, 1151A, 165C, 1243B2, 1268A, 1153C

(1-1) Week 8 (Function, Recursion)

Tutorials:

1. রিকার্শন (Recursion)
2. recursion Archives - হাসানের রাফখাতা
3. recursion and dp - smilitude
4. Recursion and Backtracking Tutorials & Notes | Basic Programming

Problems:

1. UVA: 624, 167, 539, 729, 750, 628, 10276, 11085, 574, 524, 10063, 10503, 301, 331, 193, 129, 208, 416, 861, 1262

(1-1) Week 9 (Binary Search + Others)

Tutorials:

1. Binary search implementation (Implementation variations)
2. Binary Search
3. Binary Search Practice Problems | Algorithms | page 1
4. বাইনারি সার্চ অ্যালগরিদম - Binary Search Algorithm
5. Ternary Search Tutorials & Notes | Algorithms
6. Ternary Search and its Applications
7. The great ternary search hoax

Problems:

1. Codeforces: 492B, 474B, 195B, 1221C, 1183C, 371C, 812C, 706B, 580B, 340D, 75C, 448D, 1250J, 616D, 1077D

2. LightOJ: 1043, 1072, 1307, 1138, 1088,1048, 1137, 1235,1383, 1146, 1240
3. Spoj: AGGRCOW, ACTIV
4. Codechef: MCO16503, KGP16G

(1-1) Week 10 (STL Basic)

Tutorials:

1. Power up C++ with the Standard Template Library: Part I
2. Power up C++ with the Standard Template Library: Part II: Advanced Uses
3. stl - smilitude
4. স্ট্যান্ডার্ড টেম্প্লেট লাইব্রেরী
5. C++ STL (Bangla)
6. STL

Problems:

1. UVA: 146, 10107, 514, 10935, 484, 10815, 10954, 400, 10194, 127, 540, 1203, 12058, 855, 11321, 10264, 732, 1062, 10172, 978

Level-1 Term-2

Tentative Class Schedule:

Week	Topic
1	Basic Math, Modular arithmetic, Big-mod, Modular inverse
2	Number Theory: Sieve, Prime Factorization, Number of divisors
3	Recursion, Backtracking, Tail-call recursion
4	Greedy Techniques, Task Scheduling, Kadani's Algorithm
5	Basic DP (Recursive/Iterative dp, knapsack, LIS)
Online Contest-1	
6	Advanced STL
7	Graph Basics: BFS/DFS
8	Shortest Path: Dijkstra, Floyd-Warshal, Bellman-Ford
9	Topological Sort, Tree-Diameter
10	Basic Geometry
Online Contest-2	

Guidelines:

- Participate in online contest
- Upsolve problems after attending online contest
- Try to attend and do well in onsite contest

Expectations:

- Total Number of problems solved: 300+
- CF rating:
  - -At the start: 1200+
  - -At the end: 1500+

Tips:

- Learning how to think is important at this stage. You should have gotten a feel for determining how much time you should give to each problem in order to learn properly-however, we would say do not make it less than 30 minutes, and definitely not more than 90 minutes. But make sure that you have utilized that time of thinking fully-many of us do not utilize this time fully and do not realize it. Learn to think in a IDA* approach, that is, realizing all problems have an easy solution, and can be solved in a few steps. If you find your train of thought getting too complex, or taking too many steps/difficult to visualize, then it is likely not correct, and if it is, it is most definitely not the intended solution. Learn to abandon current train of thought as soon as you realize that it may be getting too complex for you-these problems are made for humans by humans, not for Einsteins by Einsteins.
- You should have also realized that this **competitive programming business is no joke**. The future world finalists are currently working very hard, and while you are watching movies/browsing the net, they are currently solving problems at their ACM Room. You should spend most of times on thinking about problems. You can also think about problems when eating, bathing or even before sleeping
- Also, do not pick too hard problems, problems in the difficulty range where you find yourself seeking the solution almost every time. Even if you feel that you have understood the solution fully, you may not have fully internalized it-and possibly even indirectly memorize it. Ideally you must pick problems, that, if they were to appear in a contest, you can almost get the idea, but couldn't get a clear picture of the full solution within contest time.
- And at the minimum, **do every CF contests. Even if there is a CT the next day**. Solve the problems that you couldn't solve the next day. This is very, very, very important. This is called upsolving, and upsolve the next one or two problems that you couldn't solve in contest time. Perhaps leave the others for now.
- Please do look at the official solution (editorials) of problems, even if you have managed to solve one by yourself. You may find new ideas/trick there. Make a habit of learning from other people's code too.
- Look at other people's code to learn new tips/tricks after solving a problem.
- And do hide CF tags. 50% of solving a problem is about identifying the correct category of the problem, and you'll be stunted in this respect.

Expected Number of Problems solved (including previous semester)- 500 at the very least, more like 1000 if you hope for qualifying for WF in the future.

Tutorials & Problems List for Level-1 Term-2

(1-2) Week 1 (Basic Math, Modular arithmetic, Big-mod, Modular inverse)

Tutorials: (Basic Math)

1. Understanding class 1-12 math is enough
2. Euclidean Algorithm - Greatest Common Divisor
3. Lowest Common Multiple of Two Number

Problems: (Basic Math)

1. LightOJ: Basic math category
2. Codeforces: Math tag ordered by rating

Tutorials: (Modular Arithmetic & Big-Mod)

1. মডুলার অ্যারিথমেটিক
2. Introduction to Modular Arithmetic
3. Powerful tricks with calculation modulo

Problems: (Modular Arithmetic & Big-Mod)

1. LightOJ: 1067, 1054, 1102, 1419
2. UVA: 374, 10127, 128, 10176, 10212, 10489, 11029

Tutorials: (Modular Inverse)

1. Modular Multiplicative Inverse
2. Modular Inverse

Problems: (Modular Inverse)

1. UVA: 11904
2. CF: 300C, 622F, 717A, 896D

(1-2) Week 2 (Number Theory: Sieve, Prime Factorization, Number of divisors)

Tutorials: (Number Theory)

1. Category Archives: Number Theory
2. Sieve of Eratosthenes How Many Divisors of a Number
3. Generate Divisors of N (Move to 2-1)
4. A Bunch of Stuff : Number Theory in Competitive Programming [Tutorial] (Optional)

Tutorials: (Sieve)

1. প্রাইম জেনারেটর (Sieve of Eratosthenes)
2. Sieve of Eratosthenes - Generating Primes
3. Sieve of Eratosthenes

Problems: (Sieve)

1. LightOJ: 1035, 1028

Tutorials: (Bitwise Sieve)

1. বিটওয়াইজ্ সিভ(Bitwise sieve)

Problems: (Bitwise Sieve)

1. SPOJ: TDPRIMES

Tutorials: (Segmented Sieve)

2. Segmented Sieve of Eratosthenes
3. Book: Programming Contest, Data Structure and Algorithm – Mahbubul Hasan Shanto (segmented sieve topic, page-55)

Problems: (Segmented Sieve)

1. SPOJ: PRIME1, PAGAIN
2. LightOJ: 1197

Tutorials: (Prime Factorization)

1. Prime Factorization of an Integer

Problems: (Prime Factorization)

1. UVA:
    - EASY: 583, 160, 993, 10299, 11466
    - MEDIUM: 10392, 10179, 10139, 10484
    - HARD: 10622, 10680, 11347

Tutorials: (Number of Divisors)

1. Number of Divisors of an Integer

Problems: (Number of Divisiors)

1. SPOJ: COMDIV

(1-2) Week 3 (Recursion, Backtracking, Tail-call recursion)

Tutorials:

1. Attacking Recursions
2. টেইল-কল রিকার্শন অপটিমাইজেশন
3. রিকার্সিভ ফাংশনের সৌন্দর্য - ১
4. https://youtu.be/J8mChdOJWOs

Problems:

1. UVa: 624, 750, 10576, 11085, 524, 574, 10503, 193, 416, 10094

(1-2) Week 4 (Greedy Techniques, Task Scheduling, Kadani's Algorithm)

Tutorials: (Greedy)

1. Greedy is Good
2. Basics of Greedy Algorithms Tutorials & Notes | Algorithms
3. Greedy Algorithms

Problems: (Greedy)

1. BAPS Greedy and Searching Problem List
2. Easy:
    - CF - 479C
    - SPOJ – GERGOVIA
    - CF - 58B
    - UVA – 410
    - POJ - 1328
3. Medium:
    - SPOJ – MSCHED
    - CF - 313C
    - SPOJ – CMIYC
    - SPOJ – DRAGON
    - CF - 166C
    - Hackerrank - Goodland Electricity
4. Hard:
    - CF - 316A1
    - Codechef - Protecting The Poison
    - CF - 794C
    - SPOJ – TROOPS
    - A2OJ-306

Tutorials: (Task Scheduling)

1. Activity Selection Problem | Greedy Algo-1
2. Weighted Job Scheduling

3. Job Sequencing Problem
4. Page Replacement Algorithms in Operating Systems
5. Activity Selection Problem

Problems: (Task Scheduling)

1. Easy:
    - CSES - 1630
    - CSES - 1629
    - CSES - 1619
2. Medium:
    - CSES - 1164
    - Leetcode - 452
    - SPOJ - MIA14B
3. Hard:
    - CF(gym) - 101498F

Tutorials: (Kadane's Algorithm)

1. Search the subarray with the maximum/minimum sum
2. Kadane's Algorithm — (Dynamic Programming) — How and Why does it Work?
3. Largest Sum Contiguous Subarray

Problems: (Kadane's Algorithm)

1. Easy:
    - CSES – 1643
    - UVA – 507
    - UVA – 10684
    - Timus - 1146
2. Medium :
    - UVA – 787
    - CodeForces - 75D

(1-2) Week 5 (Basic DP-Recursive/Iterative dp, knapsack, LIS)

Tutorials:

1. ডাইনামিক প্রোগ্রামিং – ১ (ফিবোনাচ্চি)
2. https://www.youtube.com/playlist?list=PLrmLmBdmIlpsHaNTPP_jHHDx_os9ItYXr&app=desktop
3. Dynamic Programming: From Novice to Advanced
4. Tutorial for Dynamic Programming

5. https://www.youtube.com/playlist?list=PLl0KD3g-oDOGJUdmhFk19LaPgrfmAGQfo&app=desktop

Problems:

1. UVa - 787, 10684, 108, 11951, 10827, 111, 481, 11456, 10130, 10616, 674
2. CF - DP tag

(1-2) Week 6 (Advanced STL)

Tutorials:

1. https://ishtiaqhimu.blogspot.com/2020/10/standard-template-library-stl.html
2. Containers - C++ Reference

Problems:

1. UVa - 514, 732, 1062, 10172, 10901, 11034, 11572, 11308, 978, 11849, 1203, 11995
2. SPOJ - CTRICK
3. Timus - 1521

(1-2) Week 7 (Graph Basics: BFS/DFS)

Tutorials:

1. গ্রাফ থিওরিতে হাতেখড়ি - ১ | শাফায়েতের ব্লগ
2. Graph Editor (for graph drawing)
3. Depth First Search (DFS) (Silver) (***)
4. Breadth First Search (BFS) (Gold) (***)
5. me-shaon/bangla-programming-resources: Bangla tutorial, reference and resource list on programming topics (graph theory portion)
6. Programming Camp 2020 Session 3: Graph Theory
7. https://mehedi6022.blogspot.com/2019/02/breadth-first-search.html (BFS)
8. https://mehedi6022.blogspot.com/2020/02/0-1-bfs_11.html (0/1 bfs)
9. POTW (see Graph Theory I)
10. Video Tutorial (BAPS): https://www.youtube.com/watch?v=xswqISXxAC4&list=PLWtSipmftM8pGx7rR9xviLokw29kG_s-2&index=9
11. (Code): https://www.youtube.com/watch?v=VW85xQ6GJP4&list=PL2q4fbVm1Ik6DCzm9XZJbNwyHtHGclcEh

Problems:

1. Easy

- BFS/DFS Category(LightOJ)
- Building Roads
- USACO
- Message Route

2. Medium:
   - USACO (Flood Fill)
   - USACO
   - CSES-1668(Bicoloring), 1682 (Directed dfs)
   - USACO
   - Counting Rooms (Flood Fill)
   - Icy Perimeter (Flood Fill)
   - Where's BESSIE?(Flood Fill)
   - Why Did the Cow Cross the Road III (Flood Fill)
   - Labyrinth
   - Tree Diameter
   - Subordinates (DFS on Tree)
   - Codeforces: 862B,
   - https://csacademy.com/contest/archive/task/bfs-dfs
   - DCP-60: Holloween Party Back to All Problems (0/1 bfs)
   - Building Teams
   - Round Trip
   - Monsters
   - SPOJ.com - Problem AKBAR
3. Hard:
   - USACO
   - USACO
4. More Problems:
   - https://progvar.fun/problemsets/bfs
   - https://progvar.fun/problemsets/grids
   - https://progvar.fun/problemsets/bipartite-graphs
   - codeforces上的一个题单（原版改）
   - Problem Topics

(1-2) Week 8 (Shortest Path: Dijkstra, Floyd-Warshal, Bellman-Ford)

Tutorials:

1. গ্রাফ থিওরিতে হাতেখড়ি-৯ (ডায়াক্সট্রা)
2. গ্রাফ থিওরিতে হাতেখড়ি ১০: ফ্লয়েড ওয়ার্শল

3. গ্রাফ থিওরিতে হাতেখড়ি ১১: বেলম্যান ফোর্ড
4. Bellman-Ford Algorithm - shortest paths with negative weights
5. Dijkstra Algorithm
6. Shortest Paths with Non-Negative Edge Weights (Gold)
7. Video Tutorial with code: https://www.youtube.com/watch?v=CLnpzCnSDSY&list=PL2q4fbVm1Ik64I3VqbVGRfl_OgYzvzt9m&index=7
8. (Tushar Roy): Bellman-Ford Algorithm Single Source Shortest Path Graph Algorithm

Problems:

1. Easy:
   - CSES Problem Set - Tasks (solve Shortest Routes II to Investigation)
   - Cycle Detection
   - Shortest Path
   - Floyd - Warshall + Transitive closure
2. Medium-Hard:
   - Birthday Gift (Dijkstra)
   - Dijkstra/Floyd Warshall Category(LightOJ)
   - Bellman Ford Category(LightOJ)
   - https://progvar.fun/problemsets/shortest-paths-dags
   - https://progvar.fun/problemsets/shortest-paths-single-source
   - https://progvar.fun/problemsets/shortest-paths-single-source-negative-weights
   - ProgVar.Fun
   - https://vjudge.net/contest/341347#overview

(1-2) Week 9 (Topological Sort, Tree-Diameter)

Tutorials: (Topological Sort)

1. গ্রাফ থিওরিতে হাতেখড়ি ৭:টপোলোজিকাল সর্ট
2. Topological Sorting
3. টপোলজিকাল সর্ট - smilitude

Problems: (Topological Sort)

1. TOPOSORT - Topological Sorting
2. UVa - 10305 - Ordering Tasks
3. UVa - 124 - Following Orders
5. UVa - 200 - Rare Order
6. Problem - 510C

7. UVa - 11060 - Beverages
8. PFDEP - Project File Dependencies
9. Course Schedule
10. RPLA - Answer the boss!
11. UVa - 452 - Project Scheduling

Tutorials: (Tree Diameter)

1. ট্রি ডায়ামিটার
2. Tree Diameter — Why Does Two BFS Solution Work? | by Tarek Badr

Problems: (Tree Diameter)

1. LightOJ-1094
2. LOJ-1257
3. UVa - 10459
4. 102694A - 3 - Circumference of a Tree
5. Codeforces -102694B

(1-2) Week 10 (Basic Geometry)

Before Starting:

7. Go through and understand every theorem in secondary school (These theorems are must for solving, analyzing geometry problems). Ensure a better understanding and clear concept of vector geometry, straight lines, conics (These topics are covered in intermediate).
8. If you have done 1, go through this book " Handbook of geometry for competitive programmers "

Tutorials:

1. Programming-Problem-In-Bengali/Geometry Resources.md at master · hasancse91/Programming-Problem-In-Bengali
2. Read "Geometry Concepts" tutorials in topcoder Community - Competitive Programming - Tutorials
3. Geometry Algorithms Home
4. Basic Geometry
5. Geometry: 2D points and lines [Tutorial]
6. Geometric Algorithms

Problems:

1. Lightoj (Category - "Basic Geometry")
2. grep+: UVA problems keyword search
3. https://progvar.fun/problemsets/geometry-basics