Competitive Programming (SoC'25)

Project id - 22
Mentor - Himanshu Shete(23B0770)

Week 1 : Arrays, Time Complexity, ***Binary Search***, Two Pointers, Sorting

Theory:

(these are just resources you can always learn from youtube or other sources)

1) Arrays:
    a) https://www.geeksforgeeks.org/introduction-to-arrays-data-structure-and-algorithm-tutorials/
    b) https://www.geeksforgeeks.org/array-subarray-subsequence-and-subset/
2) Time Complexity
    a) https://www.w3schools.com/dsa/dsa_timecomplexity_theory.php
3) ***Binary Search***
    a) https://www.youtube.com/watch?v=GU7DpgHINWQ
    b) https://cp-algorithms.com/num_methods/binary_search.html
    c) Stl: binary_search(), lower_bound(),upper_bound(): https://www.geeksforgeeks.org/binary-search-functions-in-c-stl-binary_search-lower_bound-and-upper_bound/
4) Two Pointers
    a) https://www.geeksforgeeks.org/two-pointers-technique/
5) Sorting
    a) https://www.geeksforgeeks.org/sort-c-stl/
    b) Visualize:- https://visualgo.net/en/sorting
    c) https://www.w3schools.com/dsa/dsa_algo_bubblesort.php
    d) https://www.w3schools.com/dsa/dsa_algo_selectionsort.php
    e) https://www.w3schools.com/dsa/dsa_algo_insertionsort.php
    f) https://www.w3schools.com/dsa/dsa_algo_quicksort.php
    g) https://www.w3schools.com/dsa/dsa_algo_countingsort.php
    h) https://www.w3schools.com/dsa/dsa_algo_radixsort.php
    i) https://www.w3schools.com/dsa/dsa_algo_mergesort.php

Problems:

(increasing difficulty, maintain a git repo)

1. Arrays and Sorting:
    a. https://leetcode.com/problems/two-sum/description/?envType=problem-list-v2&envId=array (you can do it in o(n2))
    b. https://www.codechef.com/practice/course/arrays-strings-sorting/INTARR01/problems/DOMINANT2
    c. https://www.codechef.com/practice/course/arrays-strings-sorting/INTARR01/problems/DISTINCTCOL

    d. https://cses.fi/problemset/task/1094
    e. https://www.codechef.com/practice/course/arrays-strings-sorting/INTARR01/problems/EQUALELE
    f. https://www.codechef.com/practice/course/arrays-strings-sorting/INTARR01/problems/CFRTEST?tab=statement
    g. https://codeforces.com/problemset/problem/300/A
    h. https://www.codechef.com/practice/course/arrays-strings-sorting/INTARR01/problems/MISSP?tab=statement (bit operations needed else tle)
2. Time Complexity: (Practice observing complexity via performance on large testcases)
3. Binary Search: (i would suggest implimenting binary search every time instead of using std::binary_search() of cpp stl)
    a. https://leetcode.com/problems/binary-search/description/
    b. https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/TRICOIN
    c. https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/WAV2
    d. https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/SHKNUM
    e. https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/MINEAT (*)(this problem is beautiful go through the solution)(calculate at only the binary search locations instead of calculating everythin then binary search)
    f. https://www.codechef.com/practice/course/binary-search/INTBINS01/problems/SNAKEEAT (**)
4. Two Pointers and Sliding widow (practice more)
    a. https://www.codechef.com/practice/course/two-pointers/POINTERF/problems/PREP69?tab=statement (easy)
    b. https://www.codechef.com/practice/course/two-pointers/POINTERF/problems/PROC18A (classic)
    c. https://www.codechef.com/practice/course/two-pointers/POINTERF/problems/PREP68 (another classic)
    d. https://www.codechef.com/practice/course/two-pointers/POINTERF/problems/PREP17
    e. https://leetcode.com/problems/longest-palindromic-substring/description/?envType=problem-list-v2&envId=two-pointers
    f. https://leetcode.com/problems/container-with-most-water/description/?envType=problem-list-v2&envId=two-pointers
    g. https://leetcode.com/problems/3sum/description/?envType=problem-list-v2&envId=two-pointers (threesome)
    h. https://leetcode.com/problems/4sum/description/?envType=problem-list-v2&envId=two-pointers (foursome or k-sum)