

level two map

Session 1 : STLs

- 1-**vector** : its functions and time of each operation
- 2-**pair/tuple** : how to access and iterate over
- 3-**stack** :
 - a-find complement value (i.e balance parentheses)
 - b-monotonic stack (next greater/smaller value, histogram);
- 4-**queue/deque** : how to use. *(their usage will appear in graph and sliding window)
 - a-monotonic queue (deque) // (i.e sliding window maximum)
- 5-**set/map/unordered_map/unordered_set** :
 - a-frequency counting
 - b-find distinct elements
- 6-**priority queue** : usage, time, how to use, how to compare using more than one value

Session 2 : Prefix Sum, Two Pointers, Sliding Window

- 1-**prefix sum** :
 - a-1D sums
 - b-2D sums (range query over more than one value)
 - c-Difference array
- 2-**two pointers** :
 - a-how to deal with subarrays (i.e longest substring without repeating chars)
 - b-meet in the middle over sorted array
 - c-merge intervals
- 3-**sliding window** :
 - a-fixed-size & variable-size window

Session 3 : Binary Search

- 1-**concept**
- 2-**find value on array**
- 3-**find answer with error up to 1e-5**
- 4-**binary search on answer**
- 5-**how to use lower_bound & upper_bound (for sets as well)**

Session 4 : Bit Manipulation

- 1-**binary representation**
- 2-**operations on integers** :
 - a-set_bit b-clear_bit c-toggle_bit
- 3-**count set bits**
- 4-**bitset (STL) usage**
- 5-**subsets with bitmasks** :
 - a-intro to masks
 - b-subset concept (include exclude)
 - c-generate using iterative way
- 6-**XOR properties and operations** (i.e prefix xor)
- 7-**MITM algorithm** *(if possible)

Session 5 : Math

1-Primes

- a-primality check
- b-harmonic number / sieve
- c-SPF
- d-factorising using sieve or spf
- e-divisors

2-modular arithmetic (problems on this part focus more on gcd, co-primes, lcm) :

- a-congruence
- b-addition, multiplication
- c-gcd / LCM / extended euclidean algo
- d-division (multiplicative inverse)
- e-phi / Euler / Fermat (to find MMI)
- f-solve system with two congruences (CRT basics) *(if possible)

3-binary exponentiation with some pre-computing

4-combinatorics basics :

- a-permutations
- b-combinations
- c-distributing objects into boxes *(if possible)
- d-rule of including-excluding *(if possible)

Session 6 : Recursion & Backtracking

1-Recursion :

- a-basic concepts (Statement/Base case/Recursive call)(i.e factorial, tower of Hanoi)
- b-Fibonacci problem & how to make it fast (memoization intro)

2-backtracking

Session 7 & 8 : Graphs

1-what is graph and how to simulate it

2-Traversing DFS, BFS

3-multi-source BFS

4-some applications:

- a-connected components
- b-bipartite check
- c-path existence (how to reach from u to v)

5-what is a Tree (root, parent, max depth(diameter), DFS traversal)

6-Dijkstra

7-Floyd-Warshall

8-Cycle detection using DFS

9-Minimum Spanning Tree & DSU (prim & kruskal)

10-Topological sort (kahn's algo & DFS)

11-LCA (basic & binary lifting) *(if possible)