Competitive Programming-Beginner

Topics	Subtopics
Time/Memory Complexity	1. Importance of calculating time/memory complexity and how to do it
Basic STL (C++)	1. Vector (insert , erase , iteration)
	2. Queue
	3. Stack
	4. Deque
	1. Map (C++)
	2. Priority Queue (C++)
	3. Set (C++)
Data Structure	4. Linked list using array
	1. Bitwise operation (AND , OR , XOR and more)
	2. Manipulation of bits
Bitwise Operation	3. Some special use
	1. Calculating GCD efficiently (Euclidean algorithm)
	2. Factorization (O(√n) , O(n*ln(n)))
	3. Sieve (finding prime numbers)
	4. Bitwise sieve
	5. Prime factorization
	6. Modular Arithmetic (addition , multiplication , calculating bigmod)
	7. Fermat's little theorem and its use
	8. Totient function
	9. Inverse mod
	10. Combinatorics (factorials , counting problem)
Math	
Searching Technique	1.Linear Search
	2.Binary Search
	1. Bubble sort
	2. Insertion sort
	3. Counting sort
	4. Selection sort
	5. Quick sort
Sorting Algorithm	6. Merge sort
	1. Introduction to recursion
Recursion	2. Backtracking
Greedy and Ad-hoc	1. Introduction to greedy algorithm and ad-hoc problems
	1. What is graph theory?
	2. How to store edges? (using vector and array)
	3. How to traverse a graph? (DFS , BFS)
	4. How to solve problem using graph theory
	5. Connected Component (undirected graph)
	6. Bicoloring problem
	7. Shortest path problem (weighted and unweighted graph)
Basic Graph Theory	8. Longest path problem (tree)
	1. What is dynamic programming?
	2. State of dp and calculating time and memory complexity
	3. nCr
	4. Coin change
	5. Knapsack
	6. subset sum problem
	7. Longest Increasing subsequence (n^2)
	8. Bitmask dp
Dynamic Programming	
ouch of Advance Data Structur	1. sliding range minimum query (using deque)
	2. sparse table (where to apply: min , max, gcd and more)

Competitive Programming - Intermediate

Topic	Sub-Topic
	1. BInary search
Searching	2. Ternary search
	1. Set
	2. Map
	3. Priority Queue
	4. List
	5. Ordered Set
	6. Deque
STL	7. Bitset
	1. Binary indexed tree
	2. Segment tree

Competitive Programming - Advanced

Topics	Subtopics
	1. 1D,2D,3D Cumulative Sum
	2. 1D,2D,3D Difference Array
	3. Array Compression
	4. Bitwise Operations, Iterating over all subset efficiently
	5. Properties of exclusive or
	6. Contribution Technique
	7. Exchange Argument
	8. Union of two segment, rect
AdHoc	9. Bracket Sequence
	1. Binary Search
Search Techniques	2. Ternary Search (when it doesn't work)
	1. Vector (sort, lower_bound, upper_bound, erasing duplicates, number of occurrence of a value)
	2. Pair
	3. Set (find, erase, lower_bound, upper_bound)
	4. Map (iterating over map)
	5. Priority queue
	6. Deque , Stack, Queue -> finding the immediate small/large element, is a bracket sequence balanced
	7. Ordered Set
	8. Iterating over all possible permutation / combination
STL	9. Magic of Bitset (How does bitset really work)
	1. Depth first search (Start time, End time, back edge, forward edge, checking whether a node is in another's subtree or not, Euler Tour of a tree)
	2. Breadth first search (shortest path length, retrieving shortest path, shortest path tree, cross edge, 0-1 bfs)
	3. Articulation Point, bridge using dfs
	4. Dijkstra (shortest path dag)
	5. Floyd-Warshall (finding connectivity using bitset)
	6. Kruskal's Algorithm (Properties of minimum spanning tree)
	7. Traversing a dense graph (with bitset/set)
Graph	8. Euler Tour of a graph
	1. Hashing (Polynomial Hash, Hash of a set of integers)
	2. Trie (Can you implement a set using trie ?)
	3. String matching with bitset
	4. KMP
String	5. Aho corasick
	1. Basic Segment tree
	2. Merge Sort Tree
	3. Segment tree with Lazy Propagation
	4. Square Root Decomposition
	5. Sparse Table (min, max, and, or, gcd, lca)
	6. Disjoint Set union (Small to Large Technique)
	7. Flattening of a tree and maintaining info
Data Structure	8. Divide and Conquer Approach for trees (centroid decomposition)
	1. Divisibility
	2. Factorization (naive, O(√n))
	3. Sieve, factorization
Number Theory	4. Bezout's Identity
	1. Modular Arithmetic
	2. Factorials and Modular Inverse
	3. Stars and Bars Theorem and variation
	4. Sum in the same row/column of pascal triangle
Combinatorics	5. Basic Inclusion/Exclusion
	1. Knapsack, Different Variations (Deque, Subset Division, Bitset)
	2. Interval Dp
	3. Bitmask Dp
	4. Dp with Some Greedy Observations
Dynamic Programming	5. Optimizing memory for Dp
	1. Basic Game Theory (winning, losing state and their strategy)
	2. Nim (variations of Nim)
Game Theory	3. Grundy

	3. Sparse table			
	4. RMQ on static array			
	5. Disjoint set union			
Data Structure	6. Sqrt decomposition			
	1. Sieve			
	2. Factorization			
	3. Counting divisors			
	4. Bigmod			
	5. Modular Inverse			
	6. Totient function			
Math	7. Combinatorics			
	1. BFS, DFS			
	2. Articulation nodes			
	3. Bridges			
	4. Dijkstra			
	5. Topological sort			
	6. Floyd Warshall			
	7. MST			
	8. SCC			
Graph	9. 2 Thu			
	1. nCr			
	2. Coin change			
	3. Knapsack			
Dynamic Programming	4. Bitmask dp			
	1. KMP			
	2. Z algo			
	3. Hashing			
String:	4. Trie			
	1. Impartial games			
	2. Nim			
Games	3. Grundy			
	1. Dinic			
Flow	2. Bipartite Matching			