

L3		1. DATA STRUCTURES		2.9. Centroid Decomposition.	
Team Notebook		1.1. Union Find.		2.10. Least Common Ancestor.	
06/11/2018		1.2. Segment Tree.		2.10.1. <i>Binary Lifting.</i>	
CONTENTS		1.2.1. <i>Recursive.</i>		2.10.2. <i>Tarjan's Offline Algorithm.</i>	
1. Data Structures		1.2.2. <i>Iterative.</i>		3. STRINGS	
1.1. Union Find		1.2.3. <i>Lazy Propagation.</i>		3.1. Knuth-Morris-Pratt Algorithm.	
1.2. Segment Tree		1.2.4. <i>Persistent.</i>		3.2. Trie.	
1.3. Fenwick Tree		1.3. Fenwick Tree.		3.3. Hashing.	
1.4. Heap		1.3.1. <i>Point-updates.</i>		4. MATHEMATICS	
1.5. Balanced Binary Search Tree		1.3.2. <i>Range-updates.</i>		4.1. Binomial Coefficients.	
1.6. AVL		1.4. Heap.		4.2. Euclidean Algorithm.	
1.7. Treap		1.5. Balanced Binary Search Tree.		4.3. Primality Test.	
1.8. Sparse Table		1.6. AVL.		4.3.1. <i>Optimized Brute Force.</i>	
2. Graphs		1.7. Treap.		4.3.2. <i>Miller-Rabin.</i>	
2.1. Single-Source Shortest Paths		1.7.1. <i>Explicit.</i>		4.3.3. <i>Pollard's Rho Algorithm.</i>	
2.2. All-Pairs Shortest Paths		1.7.2. <i>Implicit.</i>		4.4. Sieve.	
2.3. Strongly Connected Components		1.8. Sparse Table.		4.4.1. <i>Sieve of Eratosthenes.</i>	
2.4. Cut Points and Bridges		2. GRAPHS		4.4.2. <i>Divisor Sieve (Modified Sieve of Eratosthenes).</i>	
2.5. Minimum Spanning Tree		2.1. Single-Source Shortest Paths.		4.4.3. <i>Phi Sieve.</i>	
2.6. Topological Sorting		2.1.1. <i>Dijkstra.</i>		4.5. Modular Exponentiation.	
2.7. Bipartite Matching		2.1.2. <i>Bellman-Ford.</i>		4.6. Modular Multiplicative Inverse.	
2.8. Maximum Flow		2.1.3. <i>A*.</i>		4.7. Chinese Remainder Theorem.	
2.9. Centroid Decomposition		2.2. All-Pairs Shortest Paths.		4.8. Numeric Integration (Simpson's Rule).	
2.10. Least Common Ancestor		2.2.1. <i>Floyd-Washall.</i>		4.9. Fast Fourier Transform.	
3. Strings		2.3. Strongly Connected Components.		4.10. Josephus Problem.	
3.1. Knuth-Morris-Pratt Algorithm		2.3.1. <i>Kosaraju.</i>		4.11. Number of Integer Points Below a Line.	
3.2. Trie		2.4. Cut Points and Bridges.		5. GEOMETRY	
3.3. Hashing		2.5. Minimum Spanning Tree.		5.1. Primitives.	
4. Mathematics		2.5.1. <i>Kruskal.</i>		5.2. Lines.	
4.1. Binomial Coefficients		2.5.2. <i>Prim.</i>		5.3. Circles.	
4.2. Euclidean Algorithm		2.6. Topological Sorting.		5.4. Polygons.	
4.3. Primality Test		2.7. Bipartite Matching.		5.5. Convex Hull (Graham's Scan).	
4.4. Sieve		2.7.1. <i>Alternating Paths Algorithm.</i>		5.6. Closest Pair of Points.	
4.5. Modular Exponentiation		2.7.2. <i>Hopcroft-Karp Algorithm.</i>		5.7. Rectilinear Minimum Spanning Tree.	
4.6. Modular Multiplicative Inverse		2.8. Maximum Flow.		6. OTHER ALGORITHMS	
4.7. Chinese Remainder Theorem		2.8.1. <i>Edmonds-Karp.</i>		6.1. 2SAT.	
4.8. Numeric Integration (Simpson's Rule)		2.8.2. <i>Dinic.</i>		6.2. Nth Permutation.	
4.9. Fast Fourier Transform					
4.10. Josephus Problem					
4.11. Number of Integer Points Below a Line					
5. Geometry					
5.1. Primitives					
5.2. Lines					
5.3. Circles					
5.4. Polygons					
5.5. Convex Hull (Graham's Scan)					
5.6. Closest Pair of Points					
5.7. Rectilinear Minimum Spanning Tree					
6. Other Algorithms					
6.1. 2SAT					
6.2. Nth Permutation					
6.3. Floyd's Cycle-Finding					
6.4. Longest Increasing Subsequence					

6.3. **Floyd's Cycle-Finding.**

6.4. **Longest Increasing Subsequence.**

