Important Links:

Math for Competitive Programming Must to do questions.

<https://www.youtube.com/watch?v=GwVDCi8CCAI>

Data structures classes link:

<https://www.cs.wcupa.edu/rkline/ds/data-structures.html>

Data Structures by Amen Goel Quora

<https://www.quora.com/What-basic-data-structures-and-algorithms-should-one-learn-before-starting-competitive-programming>

Data Structures by Sameer Gulati

<https://www.quora.com/What-is-a-list-of-data-structures-that-a-competitive-programmer-must-know>

list of important questions of leetcode // by a Chinese scholar

<https://docs.google.com/spreadsheets/d/1SbpY-04Cz8EWw3A_LBUmDEXKUMO31DBjfeMoA0dlfIA/edit#gid=126913158>

Algorithm concepts lecture by former google employee (FreeCodecamp.org)

<https://www.youtube.com/watch?v=RBSGKlAvoiM>

* ⭐️ Course Contents ⭐️ ⌨️
* ([0:00:00](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=0s)) Abstract data types ⌨️
* ([0:04:28](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=268s)) Introduction to Big-O ⌨️
* ([0:17:00](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=1020s)) Dynamic and Static Arrays ⌨️
* ([0:27:40](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=1660s)) Dynamic Array Code ⌨️
* ([0:35:03](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=2103s)) Linked Lists Introduction ⌨️
* ([0:49:16](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=2956s)) Doubly Linked List Code ⌨️
* ([0:58:26](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=3506s)) Stack Introduction ⌨️
* ([1:09:40](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=4180s)) Stack Implementation ⌨️
* ([1:12:49](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=4369s)) Stack Code ⌨️
* ([1:15:58](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=4558s)) Queue Introduction ⌨️
* ([1:22:03](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=4923s)) Queue Implementation ⌨️
* ([1:27:26](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=5246s)) Queue Code ⌨️
* ([1:31:32](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=5492s)) Priority Queue Introduction ⌨️
* ([1:44:16](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=6256s)) Priority Queue Min Heaps and Max Heaps ⌨️
* ([1:49:55](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=6595s)) Priority Queue Inserting Elements ⌨️
* ([1:59:27](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=7167s)) Priority Queue Removing Elements ⌨️
* ([2:13:00](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=7980s)) Priority Queue Code ⌨️
* ([2:28:26](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=8906s)) Union Find Introduction ⌨️
* ([2:33:57](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=9237s)) Union Find Kruskal's Algorithm ⌨️
* ([2:40:04](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=9604s)) Union Find - Union and Find Operations ⌨️
* ([2:50:30](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=10230s)) Union Find Path Compression ⌨️
* ([2:56:37](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=10597s)) Union Find Code ⌨️
* ([3:03:54](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=11034s)) Binary Search Tree Introduction ⌨️
* ([3:15:57](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=11757s)) Binary Search Tree Insertion ⌨️
* ([3:21:20](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=12080s)) Binary Search Tree Removal ⌨️
* ([3:34:47](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=12887s)) Binary Search Tree Traversals ⌨️
* ([3:46:17](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=13577s)) Binary Search Tree Code ⌨️
* ([3:59:26](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=14366s)) Hash table hash function ⌨️
* ([4:16:25](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=15385s)) Hash table separate chaining ⌨️
* ([4:24:10](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=15850s)) Hash table separate chaining source code ⌨️
* ([4:35:44](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=16544s)) Hash table open addressing ⌨️
* ([4:46:36](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=17196s)) Hash table linear probing ⌨️
* ([5:00:21](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=18021s)) Hash table quadratic probing ⌨️
* ([5:09:32](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=18572s)) Hash table double hashing ⌨️
* ([5:23:56](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=19436s)) Hash table open addressing removing ⌨️
* ([5:31:02](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=19862s)) Hash table open addressing code ⌨️
* ([5:45:36](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=20736s)) Fenwick Tree range queries ⌨️
* ([5:58:46](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=21526s)) Fenwick Tree point updates ⌨️
* ([6:03:09](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=21789s)) Fenwick Tree construction ⌨️
* ([6:09:21](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=22161s)) Fenwick tree source code ⌨️
* ([6:14:47](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=22487s)) Suffix Array introduction ⌨️
* ([6:17:54](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=22674s)) Longest Common Prefix (LCP) array ⌨️
* ([6:21:07](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=22867s)) Suffix array finding unique substrings ⌨️
* ([6:25:36](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=23136s)) Longest common substring problem suffix array ⌨️
* ([6:37:04](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=23824s)) Longest common substring problem suffix array part 2 ⌨️
* ([6:43:41](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=24221s)) Longest Repeated Substring suffix array ⌨️
* ([6:48:13](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=24493s)) Balanced binary search tree rotations ⌨️
* ([6:56:43](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=25003s)) AVL tree insertion ⌨️
* ([7:05:42](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=25542s)) AVL tree removals ⌨️
* ([7:14:12](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=26052s)) AVL tree source code ⌨️
* ([7:30:49](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=27049s)) Indexed Priority Queue | Data Structure ⌨️
* ([7:55:10](https://www.youtube.com/watch?v=RBSGKlAvoiM&t=28510s)) Indexed Priority Queue | Data Structure | Source Code

William fiset GitHub repo with all the concepts of algos

<https://github.com/williamfiset/Algorithms>