**Data Structure** (<https://www.udemy.com/course/datastructurescncpp/>)

**1. Introduction**

(1) Introduction <https://www.bilibili.com/video/BV1Ba411Y71K?p=18>

(2) Stack vs Heap Memory <https://www.bilibili.com/video/BV1Ba411Y71K?p=19>

(3) Stack vs Heap. Continued... <https://www.bilibili.com/video/BV1Ba411Y71K?p=20>

(4) Physical vs Logical Data Structures <https://www.bilibili.com/video/BV1Ba411Y71K?p=21>

(5) ADT <https://www.bilibili.com/video/BV1Ba411Y71K?p=22>

(6) Time and Space Complexity <https://www.bilibili.com/video/BV1Ba411Y71K?p=23>

(7) Time and Space Complexity from Code <https://www.bilibili.com/video/BV1Ba411Y71K?p=24>

**2. Arrays**

(1) Introduction to Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=58>

(2) Declarations of Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=59>

(3) Static vs Dynamic Arrays <https://www.bilibili.com/video/BV1Ba411Y71K?p=61>

Pointers <https://www.bilibili.com/video/BV1Ba411Y71K?p=4>

Code => <https://github.com/88happytar/dsa/blob/master/2Arrays/StaticDynamic.cpp>

(4) How to Increase Array Size <https://www.bilibili.com/video/BV1Ba411Y71K?p=63>

Code => <https://github.com/88happytar/dsa/blob/master/2Arrays/ResizeArray.cpp>

(5) 2D Arrays <https://www.bilibili.com/video/BV1Ba411Y71K?p=65>

Code => <https://github.com/88happytar/dsa/blob/master/2Arrays/2DArrays.cpp>

(6) Array Representation by Compiler <https://www.bilibili.com/video/BV1Ba411Y71K?p=67>

**Array ADT**

(1) Array ADT <https://www.bilibili.com/video/BV1Ba411Y71K?p=72>

(2) Inserting in an Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=74>

(3) Deleting from Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=76>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/ArrayInsertRemove.cpp>

(4) Linear Search <https://www.bilibili.com/video/BV1Ba411Y71K?p=78>

(5) Improving Linear Search <https://www.bilibili.com/video/BV1Ba411Y71K?p=79>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/LinearSearch.cpp>

(6) Binary Search <https://www.bilibili.com/video/BV1Ba411Y71K?p=81>

(7) Binary Search Algorithm <https://www.bilibili.com/video/BV1Ba411Y71K?p=82>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/BinarySearch.cpp>

(8) Analysis of Binary Search <https://www.bilibili.com/video/BV1Ba411Y71K?p=84>

(9) Average Case Analysis of Binary Search <https://www.bilibili.com/video/BV1Ba411Y71K?p=85>

(10) Get/Set/Avg/Max functions on Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=86>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/GetSetMaxMin.cpp>

(11) Reverse and Shift an Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=88>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/ReverseShift.cpp>

(12) Check if Array is Sorted <https://www.bilibili.com/video/BV1Ba411Y71K?p=90>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/CheckSort.cpp>

(13) Merging Arrays <https://www.bilibili.com/video/BV1Ba411Y71K?p=92>

Code => <https://github.com/88happytar/dsa/blob/master/2ArrayADT/MergeArray.cpp>

(14) Challenge: Finding Single Missing Element in an Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=99>

(15) Challenge: Finding Multiple Missing Elements in an Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=100>

(16) Challenge: Finding Missing Element in an Array Method 2 <https://www.bilibili.com/video/BV1Ba411Y71K?p=101>

(17) Challenge: Finding Duplicates in a Sorted Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=102>

(18) Challenge: Finding Duplicates in Sorted Array using Hashing <https://www.bilibili.com/video/BV1Ba411Y71K?p=103>

(19) Challenge: Finding Duplicates in an Unsorted Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=104>

(20) Challenge: Finding a Pair of Elements with sum K <https://www.bilibili.com/video/BV1Ba411Y71K?p=105>

(21) Challenge: Finding a Pair of Elements with sum K in Sorted Array <https://www.bilibili.com/video/BV1Ba411Y71K?p=106>

(22) Challenge: Finding Max and Min in a single Scan <https://www.bilibili.com/video/BV1Ba411Y71K?p=107>

**3. Linked List**

(1) About Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=149>

(2) More About Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=150>

(3) Display LinkedList <https://www.bilibili.com/video/BV1Ba411Y71K?p=151>

Code => [https://github.com/88happytar/dsa/blob/master/3LinkedList/Display.cpp](https://github.com/88happytar/dsa/blob/master/3LinkedList/DisplayAndRecursiveDisplay.cpp)

Detail explanation =>

<https://www.youtube.com/watch?v=6wXZ_m3SbEs&ab_channel=Jenny%27sLecturesCSIT>

(4) Recursive Display LinkedList <https://www.bilibili.com/video/BV1Ba411Y71K?p=152>

Code => same as above

(5) Counting Nodes in a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=155>

(6) Sum of All Elements in a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=156>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/CountAndSum.cpp>

(7) Maximum Element in a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=158>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/MaxElement.cpp>

(8) Searching in a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=160>

(9) Improve Searching in Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=161>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/Search.cpp>

(10) Inserting in a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=163>

(11) Creating a Linked List using Insert <https://www.bilibili.com/video/BV1Ba411Y71K?p=165>

(12) Creating a Linked List by Inserting at Last <https://www.bilibili.com/video/BV1Ba411Y71K?p=166>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/CreateByInsert.cpp>

(13) Inserting in a Sorted Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=167>

(14) Deleting from Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=169>

C++ Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/InsertDelete.cpp>

(15) Check if a Linked List is Sorted <https://www.bilibili.com/video/BV1Ba411Y71K?p=171>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/CheckSort.cpp>

(16) Remove Duplicates from Sorted Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=173>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/RemoveDuplicates.cpp>

(17) Reversing a Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=175>

(18) Reversing using Sliding Pointers <https://www.bilibili.com/video/BV1Ba411Y71K?p=176>

(19) Recursive Reverse for Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=177>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/Reverse.cpp>

(20) Concatenating 2 Linked Lists <https://www.bilibili.com/video/BV1Ba411Y71K?p=179>

(21) Merging 2 Linked Lists <https://www.bilibili.com/video/BV1Ba411Y71K?p=180>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/Merge.cpp>

(22) Check for LOOP in Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=182>

Code => <https://github.com/88happytar/dsa/blob/master/3LinkedList/CheckLoop.cpp>

(23) Comparison of Array with Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=2>

(24) Challenge: Finding Middle Element of a Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=3>

(25) Challenge: Finding Intersecting point of Two Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=4>

**4. Stack**

(1) Introduction to Stack <https://www.bilibili.com/video/BV1t3411M7fE?p=8>

(2) Stack using Array <https://www.bilibili.com/video/BV1t3411M7fE?p=9>

(3) Implementation of Stack using Array <https://www.bilibili.com/video/BV1t3411M7fE?p=10>

Code => <https://github.com/88happytar/dsa/blob/master/4Stack/StackWithArray.cpp>

(4) Stack using Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=12>

(5) Stack Operations using Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=13>

Code => <https://github.com/88happytar/dsa/blob/master/4Stack/StackWithLinkedList.cpp>

(6) Parenthesis Matching <https://www.bilibili.com/video/BV1t3411M7fE?p=16>

(7) Program for Parenthesis Matching <https://www.bilibili.com/video/BV1t3411M7fE?p=17>

Code => <https://github.com/88happytar/dsa/blob/master/4Stack/ParenthesisMatching.cpp>

(8) More on Parenthesis Matching <https://www.bilibili.com/video/BV1t3411M7fE?p=19>

(9) Infix to Postfix Conversion <https://www.bilibili.com/video/BV1t3411M7fE?p=20>

(10) Associativity and Unary Operators <https://www.bilibili.com/video/BV1t3411M7fE?p=21>

(11) Infix to Postfix using Stack Method 1 <https://www.bilibili.com/video/BV1t3411M7fE?p=22>

(12) Infix to Postfix using Stack Method 2 <https://www.bilibili.com/video/BV1t3411M7fE?p=23>

(13) Program for Infix to Postfix Conversion <https://www.bilibili.com/video/BV1t3411M7fE?p=24>

Code => <https://github.com/88happytar/dsa/blob/master/4Stack/InfixPostfixConversion.cpp>

(14) Challenge: Infix to Postfix with Associativity and Parenthesis <https://www.bilibili.com/video/BV1t3411M7fE?p=26>

(15) Evaluation of Postfix Expression <https://www.bilibili.com/video/BV1t3411M7fE?p=27>

(16) Program for Evaluation of Postfix <https://www.bilibili.com/video/BV1t3411M7fE?p=28>

Code => <https://github.com/88happytar/dsa/blob/master/4Stack/EvalPostfix.cpp>

**5. Queues**

(1) Queue ADT <https://www.bilibili.com/video/BV1t3411M7fE?p=30>

(2) Queue using Single Pointer <https://www.bilibili.com/video/BV1t3411M7fE?p=31>

(3) Queue using Two Pointers <https://www.bilibili.com/video/BV1t3411M7fE?p=32>

(4) Implementing Queue using Array <https://www.bilibili.com/video/BV1t3411M7fE?p=33>

Code => <https://github.com/88happytar/dsa/blob/master/5Queues/QueueWithArray.cpp>

(5) Drawback of Queue using Array <https://www.bilibili.com/video/BV1t3411M7fE?p=36>

(6) Circular Queue <https://www.bilibili.com/video/BV1t3411M7fE?p=37>

Code => <https://github.com/88happytar/dsa/blob/master/5Queues/CircularQueue.cpp>

(7) Queue using Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=39>

Code => <https://github.com/88happytar/dsa/blob/master/5Queues/QueueWithLinkedList.cpp>

(8) Double Ended Queue DEQUEUE <https://www.bilibili.com/video/BV1t3411M7fE?p=41>

(9) Priority Queues <https://www.bilibili.com/video/BV1t3411M7fE?p=42>

Code => <https://github.com/88happytar/dsa/blob/master/5Queues/PriorityQueue.cpp>

(10) Queue using 2 Stacks <https://www.bilibili.com/video/BV1t3411M7fE?p=43>

**6. Trees**

(1) Terminology <https://www.bilibili.com/video/BV1t3411M7fE?p=44>

(2) Number of Binary Trees using N Nodes <https://www.bilibili.com/video/BV1t3411M7fE?p=45>

(3) Height vs Nodes in Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=46>

(4) Internal Nodes vs External Nodes in Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=47>

(5) Strict Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=48>

(6) Height vs Node of Strict Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=49>

(7) Internal vs External Nodes of Strict Binary Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=50>

(8) n-ary Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=51>

(9) Analysis of n-Ary Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=52>

(10) Representation of Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=53>

(11) Linked Representation of Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=54>

(12) Full vs Complete Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=55>

(13) Strict vs Complete Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=56>

(14) Binary Tree Traversals <https://www.bilibili.com/video/BV1t3411M7fE?p=57>

(15) Binary Tree Traversal Easy Method 1 <https://www.bilibili.com/video/BV1t3411M7fE?p=58>

(16) Binary Tree Traversal Easy Method 2 <https://www.bilibili.com/video/BV1t3411M7fE?p=59>

(17) Binary Tree Traversal Easy Method 3 <https://www.bilibili.com/video/BV1t3411M7fE?p=60>

(18) Creating Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=61>

(19) Program to Create Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=62>

Code =>

(20) Preorder Tree Traversal <https://www.bilibili.com/video/BV1t3411M7fE?p=65>

(21) Inorder Tree Traversals Functions <https://www.bilibili.com/video/BV1t3411M7fE?p=66>

(22) Iterative Preorder <https://www.bilibili.com/video/BV1t3411M7fE?p=67>

(23) Iterative Inorder <https://www.bilibili.com/video/BV1t3411M7fE?p=68>

Code =>

(24) Level Order Traversal <https://www.bilibili.com/video/BV1t3411M7fE?p=70>

Code =>

(25) Can we Generate Tree from Traversals <https://www.bilibili.com/video/BV1t3411M7fE?p=72>

(26) Generating Tree from Traversals <https://www.bilibili.com/video/BV1t3411M7fE?p=73>

(27) Height and Count of Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=74>

Code =>

(28) Challenge: Count Leaf Nodes of a Binary Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=76>

**Binary Search Trees**

(1) BST intro <https://www.bilibili.com/video/BV1t3411M7fE?p=77>

(2) Searching in a Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=78>

(3) Inserting in a Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=79>

(4) Recursive Insert in Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=80>

(5) Creating a Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=81>

Code =>

(6) Deleting from Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=83>

Code =>

(7) Generating BST from Preorder <https://www.bilibili.com/video/BV1t3411M7fE?p=85>

(8) Program for Generating BST from Preorder <https://www.bilibili.com/video/BV1t3411M7fE?p=86>

(9) Drawbacks of Binary Search Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=87>

**AVL Trees**

(1) Introduction to AVL Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=88>

(2) Inserting in AVL with Rotations <https://www.bilibili.com/video/BV1t3411M7fE?p=89>

(3) General form of AVL Rotations <https://www.bilibili.com/video/BV1t3411M7fE?p=90>

Code =>

(4) Generating AVL Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=93>

(5) Deletion from AVL Tree with Rotations <https://www.bilibili.com/video/BV1t3411M7fE?p=94>

(6) Height Analysis of AVL Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=95>

**7. Heap**

(1) Introduction to Heap <https://www.bilibili.com/video/BV1t3411M7fE?p=105>

(2) Inserting in a Heap <https://www.bilibili.com/video/BV1t3411M7fE?p=106>

(3) Program to Insert in a Heap <https://www.bilibili.com/video/BV1t3411M7fE?p=107>

(4) Creating a Heap <https://www.bilibili.com/video/BV1t3411M7fE?p=108>

(5) Deleting from Heap and Heap Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=109>

Code =>

(6) Heapify - Faster Method for creating Heap <https://www.bilibili.com/video/BV1t3411M7fE?p=111>

(7) Heap as Priority Queue <https://www.bilibili.com/video/BV1t3411M7fE?p=112>

**8. Sorting Technique**

(1) Criteria use for Analyzing Sorts <https://www.bilibili.com/video/BV1t3411M7fE?p=113>

(2) Bubble Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=114>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/BubbleSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/BubbleSort.cpp)

(2) Insertion Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=116>

(3) Insertion Sort Continued <https://www.bilibili.com/video/BV1t3411M7fE?p=117>

(4) Program for Insertion Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=118>

(5) Analysis of Insertion Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=119>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/InsertionSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/InsertionSort.cpp)

(6) Comparing Bubble and Insertion Sorts <https://www.bilibili.com/video/BV1t3411M7fE?p=121>

(7) Selection Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=122>

(8) Program for Selection Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=123>

(9) Analysis of Selection Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=124>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/SelectionSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/SelectionSort.cpp)

(10) Idea behind Quick Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=126>

(11) Quick Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=127>

(12) Analysis of Quick Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=128>

(13) Analysis of Quick Sort Continued <https://www.bilibili.com/video/BV1t3411M7fE?p=129>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/QuickSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/QuickSort.cpp)

(14) Merging <https://www.bilibili.com/video/BV1t3411M7fE?p=131>

(15) Iterative Merge Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=132>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/MergeSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/MergeSort.cpp)

(16) Recursive Merge Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=134>

Code =>

(17) Count Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=136>

Code => [https://github.com/88happytar/dsa/blob/master/8Sort/CountSort.cpp](https://github.com/88happytar/dsa/blob/master/5sort/CountSort.cpp)

(18) Bin Bucket Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=138>

(19) Radix Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=139>

(20) Shell Sort <https://www.bilibili.com/video/BV1t3411M7fE?p=140>

Code =>

**9. Hashing Technique**

(1) Introduction to Hashing <https://www.bilibili.com/video/BV1t3411M7fE?p=142>

(2) Chaining <https://www.bilibili.com/video/BV1t3411M7fE?p=143>

Code => <https://github.com/88happytar/dsa/blob/master/9Hashing/Chaining.cpp>

(3) Linear Probing <https://www.bilibili.com/video/BV1t3411M7fE?p=145>

Code => <https://github.com/88happytar/dsa/blob/master/9Hashing/LinearProbing.cpp>

(4) Quadratic Probing <https://www.bilibili.com/video/BV1t3411M7fE?p=147>

(5) Double Hashing <https://www.bilibili.com/video/BV1t3411M7fE?p=148>

(6) Hash Function Ideas <https://www.bilibili.com/video/BV1t3411M7fE?p=149>

**10. Graphs**

(1) Introduction to Graphs <https://www.bilibili.com/video/BV1t3411M7fE?p=150>

(2) Representation of Undirected Graph <https://www.bilibili.com/video/BV1t3411M7fE?p=151>

(3) Representation of Directed Graphs <https://www.bilibili.com/video/BV1t3411M7fE?p=152>

(4) Breadth First Search <https://www.bilibili.com/video/BV1t3411M7fE?p=153>

(5) Program for BFS <https://www.bilibili.com/video/BV1t3411M7fE?p=154>

(6) Depth First Search <https://www.bilibili.com/video/BV1t3411M7fE?p=155>

(7) Program for DFS <https://www.bilibili.com/video/BV1t3411M7fE?p=156>

Code =>

(8) Spanning Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=158>

(9) Prim's Minimum Cost Spanning Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=159>

(10) Prim's Program <https://www.bilibili.com/video/BV1t3411M7fE?p=160>

Code =>

(11) Kruskal's Minimum Cost Spanning Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=162>

(12) Disjoint Subsets <https://www.bilibili.com/video/BV1t3411M7fE?p=163>

(13) Kruskal's Program <https://www.bilibili.com/video/BV1t3411M7fE?p=164>

Code =>

**11. Dynamic Programming**

(1) Dynamic Programming Introduction <https://www.bilibili.com/video/BV1ht411v7Ku?p=77>

(2) Memoization 1 <https://www.bilibili.com/video/BV1ht411v7Ku?p=78>

(3) Memoization 2 <https://www.bilibili.com/video/BV1ht411v7Ku?p=79>

(4) Fibonacci and Dynamic Programming <https://www.bilibili.com/video/BV1ht411v7Ku?p=80>

(5) Dynamic Programming <https://www.bilibili.com/video/BV1ht411v7Ku?p=81>

(6) Implementing Dynamic Programming <https://www.bilibili.com/video/BV1ht411v7Ku?p=82>

**12. Matrix**

(1)

(2)

**13. Linked List** **PART 2**

(23) Circular Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=185>

(24) Display Circular Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=186>

Code =>

(25) Inserting in a Circular Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=188>

Code =>

(26) Deleting From Circular Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=190>

Code =>

(27) Doubly Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=192>

Code =>

(28) Insert in a Doubly Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=194>

Code =>

(29) Deleting from Doubly Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=196>

Code =>

(30) Reverse a Doubly Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=198>

Code =>

(31) Circular Doubly Linked List <https://www.bilibili.com/video/BV1Ba411Y71K?p=200>

(32) Comparison of Linked List <https://www.bilibili.com/video/BV1t3411M7fE?p=1>

14. **Trees PART 2**

(1) 2-3 Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=96>

(2) 2-3-4 Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=97>

(3) Red-Black Trees Introduction <https://www.bilibili.com/video/BV1t3411M7fE?p=98>

(4) Red-Black Tree creation <https://www.bilibili.com/video/BV1t3411M7fE?p=99>

(5) Red-Black Trees vs 2-3-4 Trees <https://www.bilibili.com/video/BV1t3411M7fE?p=100>

(6) Creating Red-Black Tree similar to Creating 2-3-4 Tree <https://www.bilibili.com/video/BV1t3411M7fE?p=101>

(7) Red-Black Tree Deletion Cases <https://www.bilibili.com/video/BV1t3411M7fE?p=102>

(8) Red-Black Tree Deletion Examples <https://www.bilibili.com/video/BV1t3411M7fE?p=103>

(9) Red-Black Tree vs 2-3-4 Tree Deletion <https://www.bilibili.com/video/BV1t3411M7fE?p=104>