|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Program Code** | **Course Title**  **Problem Solving with Data Structures & Algorithms** | **L** | **T** | **P** | **C** | **CH** | **Course**  **Type** |
|  | **Course Code** | **0** | **0** | **1** | **0** | **35** | **VAC** |
| **Pre-requisite** | | **-** | | | | | | |
| **Co-requisite** | | **-** | | | | | | |
| **Anti-requisite** | | **-** | | | | | | |

**a. Course Description**

This course covers the design, analysis, and implementation of basic data structures in C++. Algorithms operating on the data structures are also covered. It is fundamentally a course about data structures, not about C++. Then, we delve deeper into the design, analysis and implementation of such data structures. Asymptotic analysis of algorithms and data structures is discussed. Other C++ features needed for generic implementations of the data structures are introduced along the way.

**b. Course Objectives**

The course is designed to develop skills to design and analyze simple linear and non linear data structures. It strengthen the ability to the students to identify and apply the suitable data structure for the given real world problem. It enables them to gain knowledge in practical applications of data structures.

**c. Course Outcomes**

|  |  |
| --- | --- |
| **CO1** | To make the students familiar with logic building for different Data structure concepts. |
| **CO2** | To improve the problem solving capabilities on Hackerrank Platform. |
| **CO3** | To get proficiency in developing and implementing efficient solutions for different problems |
| **CO4** | Apply important algorithmic design paradigms and methods of analysis. |
| **CO5** | Analyze the asymptotic performance of algorithms. |

**d. Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Lab** | **Topic** | **Syllabus** |
| **Week 1** | Module 1 | **Arrays** | 1. <https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/minimum-additions-0142ac80/>  2. <https://www.hackerearth.com/practice/data-structures/arrays/multi-dimensional/practice-problems/algorithm/largest-square-3d7a938a/>  3. https://leetcode.com/problems/burst-balloons/ |
| Module 2 | 1. <https://leetcode.com/problems/container-with-most-water/>  2. <https://leetcode.com/problems/next-permutation/>  3. https://leetcode.com/problems/first-missing-positive/ |
| Module 3 | 1. <https://leetcode.com/problems/remove-duplicates-from-sorted-array/>  2. <https://leetcode.com/problems/search-in-rotated-sorted-array/>  3. https://leetcode.com/problems/trapping-rain-water/ |
| Module 4 | 1. <https://leetcode.com/problems/search-insert-position/>  2. <https://leetcode.com/problems/jump-game/>  3. https://leetcode.com/problems/n-queens/ |
| Module 5 | 1. <https://leetcode.com/problems/maximum-subarray/>  2. <https://leetcode.com/problems/word-search/>  3. https://leetcode.com/problems/maximal-rectangle/ |
| **Week 2** | Module 6 | **Stacks & Queues** | 1. <https://leetcode.com/problems/valid-parentheses/>  2. <https://leetcode.com/problems/simplify-path/>  3. <https://leetcode.com/problems/sliding-window-maximum/> |
| Module 7 | 1. <https://leetcode.com/problems/implement-queue-using-stacks/>  2. <https://leetcode.com/problems/reorder-list/>  3. https://leetcode.com/problems/basic-calculator/ |
| Module 8 | 1. <https://leetcode.com/problems/min-stack/>  2. <https://leetcode.com/problems/design-circular-queue/>  3. https://leetcode.com/problems/create-maximum-number/ |
| Module 9 | 1. <https://leetcode.com/problems/number-of-students-unable-to-eat-lunch/>  2. <https://leetcode.com/problems/product-of-the-last-k-numbers/>  3. https://leetcode.com/problems/longest-valid-parentheses/ |
| Module 10 | 1. <https://leetcode.com/problems/time-needed-to-buy-tickets/>  2. <https://leetcode.com/problems/valid-parenthesis-string/>  3. https://leetcode.com/problems/number-of-atoms/ |
| **Week 3** | Module 11 | **Tress & Heaps** | 1. <https://www.hackerrank.com/challenges/tree-preorder-traversal/problem>  2. <https://www.hackerrank.com/challenges/swap-nodes-algo/problem>  3. https://leetcode.com/problems/maximum-depth-of-binary-tree/ |
| Module 12 | 1. <https://www.hackerrank.com/challenges/tree-postorder-traversal/problem>  2. <https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/>  3. https://www.codechef.com/problems/MINMAX3 |
| Module 13 | 1. <https://www.hackerrank.com/challenges/tree-top-view/problem>  2. <https://www.codechef.com/problems/GRUSH>  3. https://leetcode.com/problems/trim-a-binary-search-tree/ |
| Module 14 | 1. <https://www.hackerrank.com/challenges/tree-level-order-traversal/problem>  2. <https://leetcode.com/problems/super-ugly-number/>  3. https://leetcode.com/problems/sort-characters-by-frequency/ |
| Module 15 | 1. <https://leetcode.com/problems/increasing-order-search-tree/>  2. <https://www.hackerearth.com/practice/data-structures/trees/binary-and-nary-trees/practice-problems/algorithm/magical-tree-1-e7f8cabd/>  3. https://leetcode.com/problems/strong-password-checker/ |
| **Week 4** | Module 16 | **Linked List** | 1. <https://leetcode.com/problems/merge-two-sorted-lists/>  2. <https://leetcode.com/problems/swap-nodes-in-pairs/>  3. https://leetcode.com/problems/merge-k-sorted-lists/ |
| Module 17 | 1. <https://leetcode.com/problems/linked-list-cycle/>  2. <https://leetcode.com/problems/linked-list-cycle-ii/>  3. https://leetcode.com/problems/design-skiplist/ |
| Module 18 | 1. https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer/  2. https://leetcode.com/problems/linked-list-components/  3. https://leetcode.com/problems/reverse-nodes-in-k-group/ |
| Module 19 | 1. <https://leetcode.com/problems/intersection-of-two-linked-lists/>  2. <https://leetcode.com/problems/odd-even-linked-list/>  3. https://leetcode.com/problems/design-skiplist/ |
| Module 20 | 1. https://leetcode.com/problems/remove-linked-list-elements/  2. https://leetcode.com/problems/linked-list-random-node/  3. https://leetcode.com/problems/lfu-cache/ |
| **Week 5** | Module 21 | **Searching & Sorting** | 1. <https://leetcode.com/problems/merge-sorted-array/>  2. https://leetcode.com/problems/merge-intervals/  3. <https://leetcode.com/problems/count-of-range-sum/> |
| Module 22 | 1. <https://leetcode.com/problems/intersection-of-two-arrays/>  2. <https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/>  3. https://www.hackerrank.com/challenges/cloudy-day/problem |
| Module 23 | 1. <https://leetcode.com/problems/majority-element/>  2. <https://leetcode.com/problems/single-element-in-a-sorted-array/>  3. https://leetcode.com/problems/split-array-largest-sum/ |
| Module 24 | 1. <https://leetcode.com/problems/find-the-difference/>  2. <https://leetcode.com/problems/largest-divisible-subset/>  3. https://leetcode.com/problems/sort-an-array/ |
| Module 25 | 1. <https://leetcode.com/problems/guess-number-higher-or-lower/>  2. <https://leetcode.com/problems/top-k-frequent-words/>  3. https://leetcode.com/problems/maximum-gap/ |
| **Week 6** | Module 26 | **DFS & BFS** | 1. <https://leetcode.com/problems/same-tree/>  2. <https://leetcode.com/problems/water-and-jug-problem/>  3. https://leetcode.com/problems/freedom-trail/ |
| Module 27 | 1. <https://leetcode.com/problems/path-sum/>  2. <https://leetcode.com/problems/most-frequent-subtree-sum/>  3. https://leetcode.com/problems/word-ladder/ |
| Module 28 | 1. <https://leetcode.com/problems/sum-of-left-leaves/>  2. <https://leetcode.com/problems/array-nesting/>  3. https://leetcode.com/problems/contain-virus/ |
| Module 29 | 1. <https://leetcode.com/problems/invert-binary-tree/>  2. <https://leetcode.com/problems/find-bottom-left-tree-value/>  3. https://leetcode.com/problems/longest-increasing-path-in-a-matrix/ |
| Module 30 | 1. <https://leetcode.com/problems/island-perimeter/>  2. <https://leetcode.com/problems/evaluate-division/>  3. https://leetcode.com/problems/cracking-the-safe/ |
| **Week 7** | Module 31 | **Graph Theory** | 1. <https://leetcode.com/problems/find-the-town-judge/>  2. <https://leetcode.com/problems/clone-graph/>  3. https://leetcode.com/problems/redundant-connection-ii/ |
| Module 32 | 1. <https://leetcode.com/problems/find-center-of-star-graph/>  2. <https://leetcode.com/problems/course-schedule/>  3. https://leetcode.com/problems/couples-holding-hands/ |
| Module 33 | 1. <https://leetcode.com/problems/find-if-path-exists-in-graph/>  2. <https://leetcode.com/problems/minimum-height-trees/>  3. https://leetcode.com/problems/sum-of-distances-in-tree/ |
| Module 34 | 1.https://www.hackerrank.com/challenges/bfsshortreach/problem?isFullScreen=true  2. <https://leetcode.com/problems/network-delay-time/>  3. https://leetcode.com/problems/shortest-path-visiting-all-nodes/ |
| Module 35 | 1. <https://www.hackerrank.com/challenges/journey-to-the-moon/problem?isFullScreen=true>  2. <https://leetcode.com/problems/all-paths-from-source-to-target/>  3. https://leetcode.com/problems/critical-connections-in-a-network/ |
| **Week 8** | Module 36 | **Dynamic Programming & Greedy Method** | 1. <https://leetcode.com/problems/climbing-stairs/>  2. <https://leetcode.com/problems/generate-parentheses/>  3. https://leetcode.com/problems/candy/ |
| Module 37 | 1. <https://leetcode.com/problems/longest-palindrome/>  2. <https://leetcode.com/problems/minimum-path-sum/>  3. https://leetcode.com/problems/edit-distance/ |
| Module 38 | 1. <https://leetcode.com/problems/pascals-triangle/>  2. <https://leetcode.com/problems/integer-replacement/>  3. https://leetcode.com/problems/number-of-digit-one/ |
| Module 39 | 1. <https://leetcode.com/problems/assign-cookies/>  2. <https://leetcode.com/problems/unique-binary-search-trees/>  3. https://leetcode.com/problems/minimum-cost-to-hire-k-workers/ |
| Module 40 | 1. <https://leetcode.com/problems/lemonade-change/>  2. <https://leetcode.com/problems/triangle/>  3. https://leetcode.com/problems/minimum-number-of-refueling-stops/ |

**e. Text Books:**

1. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.
2. Data Structures using C & C++ -By Ten Baum Publisher – Prentice-Hall International.
3. Fundamentals of Computer Algorithms by Horowitz, Sahni, Galgotia Pub. 2001 ed.

**f. Reference Books:**

1. Fundamentals of Data Structures in C++-By Sartaj Sahani.
2. Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher-Thomson Learning.