Data Structures & Algorithms made easy

The Winter of Code  is a program aimed to increase participation for the Google Summer of **Code** program among students in colleges and universities. .

The project aims to contribute to the repository of NeoAlgo in TesseractCoding organisation consisting of Data Structures and Algorithms in different languages of Computer Science. The objective is:

1) To bring all possible Data Structures and Algorithms under one roof in different languages of computer science.

2) With the aid and assistance of different possible Data Structures and Algorithms, the coding will be done in such a fashion that the code will be optimised and efficient in terms of time and space complexity. Alongside, adding comments to the codes is one of the priority as it would eventually improve the Readibility of the program.

3) Documenting the code along with the output is one of the essential aim and henceforth contributing to the open source.

1.2 Feasibility & Scope

Given the largest scope of the project and the diverse ambit of the Data Structures and Algorithms,it becomes critical and crucial to desribe the boundaries of the stuff which are

-DSA is constantly evolving:

Implementation is not that difficult , work is easy to implement but the main point is designing is where it takes time. We need to use optimum algorithm or the limited available resources.

This is main fulcrum majority of the companies hire engineers in this way. It allows us to use available resources in an optimized fashion.

-Technical Details:

Using Data Structures and Algorithms, it becomes fairly and squarely easy. But coding in an efficient and optimised fashion using DSA with appropriate space and time complexity isn't an easy thing to do.

This specific approach was chosen because of the following reasons:

1)The program should be maintainable

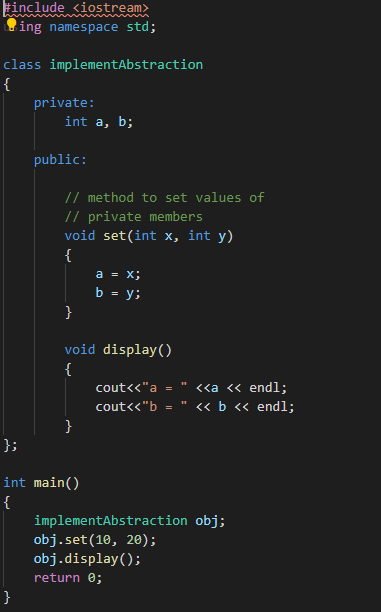
The most standard Data Structures and Algorithms will be added to the repository of NeoAlgo in the language of . It offers flexibility.

2)Modularity and Proper Object Oriented Programming will be strictly followed in order to code according to the standard practices. Optimised and Efficient Approach will be followed with efficient time and space complexity.

3.Syntax Analyser & Semantic Analyser will be shown.

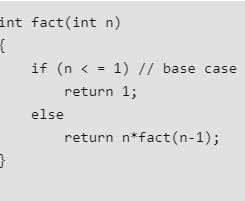
For instance: Following oops

Abstraction using access specifiers



4) Design: programs in accordance to standard Data Structures and Algorithms will be constructed according to the design. Design implies coding will be done with pre-thought process in order to evade errors as thoughts are cheaper than debugging.

5) Elegance: Sounds weird when it comes to coding or programming but elegance is essentially Beauty in the code. Stuff like following proper indentation in the code coupled with simplicity and Effeciency will be followed while contributing



I used Algorithms by Sedgwick (Moderate) Data Structures and Algorithms made easy by Narasimha Karumanchi as a reference to while coding the data structures and algorithms . Also Love Babbar’s repository.

6)Using Standard Template Library (STL) : STL is apparently powerful template library and template functions that implements many common Data Structures and Algorithms easily, removing the chunk of complexity.

7)Object Oriented Programming

It's faster and easier to execute. One of the lethal advantage of usage of OOPS is that it follows the approach of DRY(Don't Repeat yourself). We can extract the chunk of codes for the common application.

COMMUNITY BONDING

Get in touch with my project mentor, introduce myself, workout a good time to communicate (taking into account timezone differences and their schedule).

Ask the mentor for any suggestions/changes to the project. The project proposal reflects my knowledge as of writing, there may be essential amendments neccessary and it would be better to make them early.

Go through the NeoAlgo code base again. Make notes of which tokens: - datatypes, identifiers, punctuation marks, keywords etc are used most often. Some of them may be functions defined only inside certain libraries, pay special attention to those.

Go through all the example programs given in the Github Repository, the inbuilt examples are what we'll be using as a template for a basic Data Structures and Algorithms program, so it is essential to get familiar with them.

Get in touch with other WOCer's .

PHASE 1: 10th December - 20th December

Majority of the languages in the repository such as Python and C++ has got major chunk of DSA covered. However the repositories of languages such as C-Sharp and Go doesn't have much of the DSA part covered. Although repositories in languages such as Python and C++ got the DSA but the variety of questions are missing. My work will enhance the repository by adding more questions on different DSA. Moreover advanced DSA are missing which are critical and crucial for competitive programming. Also documentation will be done.

**ARRAY**

Reverse the array

Find the maximum and minimum element in an array

Find the "Kth" max and min element of an array

Given an array which consists of only 0, 1 and 2. Sort the array without using any sorting algo

Move all the negative elements to one side of the array

Find the Union and Intersection of the two sorted arrays.

Write a program to cyclically rotate an array by one.

find Largest sum contiguous Subarray [V. IMP]

Minimise the maximum difference between heights [V.IMP]

Minimum no. of Jumps to reach end of an array

find duplicate in an array of N+1 Integers

Merge 2 sorted arrays without using Extra space.

Kadane's Algo [V.V.V.V.V IMP]

Merge Intervals

Next Permutation

Count Inversion

Best time to buy and Sell stock

find all pairs on integer array whose sum is equal to given number

find common elements In 3 sorted arrays

Rearrange the array in alternating positive and negative items with O(1) extra space

Find if there is any subarray with sum equal to 0

Find factorial of a large number

find maximum product subarray

Find longest coinsecutive subsequence

Given an array of size n and a number k, fin all elements that appear more than " n/k " times.

Maximum profit by buying and selling a share atmost twice

Find whether an array is a subset of another array

Find the triplet that sum to a given value

Trapping Rain water problem

Chocolate Distribution problem

Smallest Subarray with sum greater than a given value

Three way partitioning of an array around a given value

Minimum swaps required bring elements less equal K together

Minimum no. of operations required to make an array palindrome

Median of 2 sorted arrays of equal size

Median of 2 sorted arrays of different size

Find first and last positions of an element in a sorted array

Find a Fixed Point (Value equal to index) in a given array

Search in a rotated sorted array

square root of an integer

Maximum and minimum of an array using minimum number of comparisons

Optimum location of point to minimize total distance

Find the repeating and the missing

find majority element

Searching in an array where adjacent differ by at most k

find a pair with a given difference

find four elements that sum to a given value

maximum sum such that no 2 elements are adjacent

Count triplet with sum smaller than a given value

merge 2 sorted arrays

print all subarrays with 0 sum

Product array Puzzle

Sort array according to count of set bits

minimum no. of swaps required to sort the array

Bishu and Soldiers

Rasta and Kheshtak

Kth smallest number again

Find pivot element in a sorted array

K-th Element of Two Sorted Arrays

Aggressive cows

Book Allocation Problem

EKOSPOJ:

Job Scheduling Algo

Missing Number in AP

Smallest number with atleastn trailing zeroes infactorial

Painters Partition Problem:

ROTI-Prata SPOJ

DoubleHelix SPOJ

Subset Sums

Findthe inversion count

Implement Merge-sort in-place

Partitioning and Sorting Arrays with Many Repeated Entries

**LINKED LIST**

Write a Program to reverse the Linked List. (Both Iterative and recursive)

Reverse a Linked List in group of Given Size. [Very Imp]

Write a program to Detect loop in a linked list.

Write a program to Delete loop in a linked list.

Find the starting point of the loop.

Remove Duplicates in a sorted Linked List.

Remove Duplicates in a Un-sorted Linked List.

Write a Program to Move the last element to Front in a Linked List.

Add “1” to a number represented as a Linked List.

Add two numbers represented by linked lists.

Intersection of two Sorted Linked List.

Intersection Point of two Linked Lists.

Merge Sort For Linked lists.[Very Important]

Quicksort for Linked Lists.[Very Important]

Find the middle Element of a linked list.

Check if a linked list is a circular linked list.

Split a Circular linked list into two halves.

Write a Program to check whether the Singly Linked list is a palindrome or not.

Deletion from a Circular Linked List.

Reverse a Doubly Linked list.

Find pairs with a given sum in a DLL.

Count triplets in a sorted DLL whose sum is equal to given value “X”.

Sort a “k”sorted Doubly Linked list.[Very IMP]

Rotate DoublyLinked list by N nodes.

Rotate a Doubly Linked list in group of Given Size.[Very IMP]

Can we reverse a linked list in less than O(n) ?

Why Quicksort is preferred for. Arrays and Merge Sort for LinkedLists ?

Flatten a Linked List

Sort a LL of 0's, 1's and 2's

Clone a linked list with next and random pointer

Merge K sorted Linked list

Multiply 2 no. represented by LL

Delete nodes which have a greater value on right side

Segregate even and odd nodes in a Linked List

Program for n’th node from the end of a Linked List

Find the first non-repeating character from a stream of characters

**TREE**

level order traversal

Reverse Level Order traversal

Height of a tree

Diameter of a tree

Mirror of a tree

Inorder Traversal of a tree both using recursion and Iteration

Preorder Traversal of a tree both using recursion and Iteration

Postorder Traversal of a tree both using recursion and Iteration

Left View of a tree

Right View of Tree

Top View of a tree

Bottom View of a tree

Zig-Zag traversal of a binary tree

Check if a tree is balanced or not

Diagnol Traversal of a Binary tree

Boundary traversal of a Binary tree

Construct Binary Tree from String with Bracket Representation

Convert Binary tree into Doubly Linked List

Convert Binary tree into Sum tree

Construct Binary tree from Inorder and preorder traversal

Find minimum swaps required to convert a Binary tree into BST

Check if Binary tree is Sum tree or not

Check if all leaf nodes are at same level or not

Check if a Binary Tree contains duplicate subtrees of size 2 or more [ IMP ]

Check if 2 trees are mirror or not

Sum of Nodes on the Longest path from root to leaf node

Check if given graph is tree or not. [ IMP ]

Find Largest subtree sum in a tree

Maximum Sum of nodes in Binary tree such that no two are adjacent

Print all "K" Sum paths in a Binary tree

Find LCA in a Binary tree

Find distance between 2 nodes in a Binary tree

Kth Ancestor of node in a Binary tree

Find all Duplicate subtrees in a Binary tree [ IMP ]

Tree Isomorphism Problem

Fina a value in a BST

Deletion of a node in a BST

Find min and max value in a BST

Find inorder successor and inorder predecessor in a BST

Check if a tree is a BST or not

Populate Inorder successor of all nodes

Find LCA of 2 nodes in a BST

Construct BST from preorder traversal

Convert Binary tree into BST

Convert a normal BST into a Balanced BST

Merge two BST [ V.V.V>IMP ]

Find Kth largest element in a BST

Find Kth smallest element in a BST

Count pairs from 2 BST whose sum is equal to given value "X"

Find the median of BST in O(n) time and O(1) space

Count BST ndoes that lie in a given range

Replace every element with the least greater element on its right

Given "n" appointments, find the conflicting appointments

Check preorder is valid or not

Check whether BST contains Dead end

Largest BST in a Binary Tree [ V.V.V.V.V IMP ]

Flatten BST to sorted list

**PHASE 2:**

**GREEDY ALGORITHM**

Activity Selection Problem

Job SequencingProblem

Huffman Coding

Water Connection Problem

Fractional Knapsack Problem

Greedy Algorithm to find Minimum number of Coins

Maximum trains for which stoppage can be provided

Minimum Platforms Problem

Buy Maximum Stocks if i stocks can be bought on i-th day

Find the minimum and maximum amount to buy all N candies

Minimize Cash Flow among a given set of friends who have borrowed money from each other

Minimum Cost to cut a board into squares

Check if it is possible to survive on Island

Find maximum meetings in one room

Maximum product subset of an array

Maximize array sum after K negations

Maximize the sum of arr[i]\*i

Maximum sum of absolute difference of an array

Maximize sum of consecutive differences in a circular array

Minimum sum of absolute difference of pairs of two arrays

Program for Shortest Job First (or SJF) CPU Scheduling

Program for Least Recently Used (LRU) Page Replacement algorithm

Smallest subset with sum greater than all other elements

Chocolate Distribution Problem

DEFKIN -Defense of a Kingdom

DIEHARD -DIE HARD

GERGOVIA -Wine trading in Gergovia

Picking Up Chicks

CHOCOLA –Chocolate

ARRANGE -Arranging Amplifiers

K Centers Problem

Minimum Cost of ropes

Find smallest number with given number of digits and sum of digits

Rearrange characters in a string such that no two adjacent are same

Find maximum sum possible equal sum of three stacks

Rat in a maze Problem

Printing all solutions in N-Queen Problem

Word Break Problem using Backtracking

Remove Invalid Parentheses

Sudoku Solver

m Coloring Problem

Print all palindromic partitions of a string

Subset Sum Problem

The Knight’s tour problem

Tug of War

Find shortest safe route in a path with landmines

Combinational Sum

Find Maximum number possible by doing at-most K swaps

Print all permutations of a string

Find if there is a path of more than k length from a source

Longest Possible Route in a Matrix with Hurdles

Print all possible paths from top left to bottom right of a mXn matrix

Partition of a set intoK subsets with equal sum

Find the K-th Permutation Sequence of first N natural numbers

**STACK,QUEUE & HEAP**

Implement Stack from Scratch

Implement Queue from Scratch

Implement 2 stack in an array

find the middle element of a stack

Implement "N" stacks in an Array

Check the expression has valid or Balanced parenthesis or not.

Reverse a String using Stack

Design a Stack that supports getMin() in O(1) time and O(1) extra space.

Find the next Greater element

The celebrity Problem

Arithmetic Expression evaluation

Evaluation of Postfix expression

Implement a method to insert an element at its bottom without using any other data structure.

Reverse a stack using recursion

Sort a Stack using recursion

Merge Overlapping Intervals

Largest rectangular Area in Histogram

Length of the Longest Valid Substring

Expression contains redundant bracket or not

Implement Stack using Queue

Implement Stack using Deque

Stack Permutations (Check if an array is stack permutation of other)

Implement Queue using Stack

Implement "n" queue in an array

Implement a Circular queue

LRU Cache Implementationa

Reverse a Queue using recursion

Reverse the first “K” elements of a queue

Interleave the first half of the queue with second half

Find the first circular tour that visits all Petrol Pumps

Minimum time required to rot all oranges

Distance of nearest cell having 1 in a binary matrix

First negative integer in every window of size “k”

Check if all levels of two trees are anagrams or not.

Sum of minimum and maximum elements of all subarrays of size “k”.

Minimum sum of squares of character counts in a given string after removing “k” characters.

Queue based approach or first non-repeating character in a stream.

Next Smaller Element

Implement a Maxheap/MinHeap using arrays and recursion.

Sort an Array using heap. (HeapSort)

Maximum of all subarrays of size k.

“k” largest element in an array

Kth smallest and largest element in an unsorted array

Merge “K” sorted arrays. [ IMP ]

Merge 2 Binary Max Heaps

Kth largest sum continuous subarrays

Leetcode- reorganize strings

Merge “K” Sorted Linked Lists [V.IMP]

Smallest range in “K” Lists

Median in a stream of Integers

Check if a Binary Tree is Heap

Connect “n” ropes with minimum cost

Convert BST to Min Heap

Convert min heap to max heap

Rearrange characters in a string such that no two adjacent are same.

Minimum sum of two numbers formed from digits of an array

**GRAPH & DP**

Create a Graph, print it

Implement BFS algorithm

Implement DFS Algo

Detect Cycle in Directed Graph using BFS/DFS Algo

Detect Cycle in UnDirected Graph using BFS/DFS Algo

Search in a Maze

Minimum Step by Knight

flood fill algo

Clone a graph

Making wired Connections

word Ladder

Dijkstra algo

Implement Topological Sort

Minimum time taken by each job to be completed given by a Directed Acyclic Graph

Find whether it is possible to finish all tasks or not from given dependencies

Find the no. of Isalnds

Given a sorted Dictionary of an Alien Language, find order of characters

Implement Kruksal’sAlgorithm

Implement Prim’s Algorithm

Total no. of Spanning tree in a graph

Implement Bellman Ford Algorithm

Implement Floyd warshallAlgorithm

Travelling Salesman Problem

Graph ColouringProblem

Snake and Ladders Problem

Find bridge in a graph

Count Strongly connected Components(Kosaraju Algo)

Check whether a graph is Bipartite or Not

Detect Negative cycle in a graph

Longest path in a Directed Acyclic Graph

Journey to the Moon

Cheapest Flights Within K Stops

Oliver and the Game

Water Jug problem using BFS

Water Jug problem using BFS

Find if there is a path of more thank length from a source

M-ColouringProblem

Minimum edges to reverse o make path from source to destination

Paths to travel each nodes using each edge(Seven Bridges)

Vertex Cover Problem

Chinese Postman or Route Inspection

Number of Triangles in a Directed and Undirected Graph

Minimise the cashflow among a given set of friends who have borrowed money from each other

Two Clique Problem

Construct a trie from scratch

Find shortest unique prefix for every word in a given list

Word Break Problem | (Trie solution)

Given a sequence of words, print all anagrams together

Implement a Phone Directory

Print unique rows in a given boolean matrix

Coin ChangeProblem

Knapsack Problem

Binomial CoefficientProblem

Permutation CoefficientProblem

Program for nth Catalan Number

Matrix Chain Multiplication

Edit Distance

Subset Sum Problem

Friends Pairing Problem

Gold Mine Problem

Assembly Line SchedulingProblem

Painting the Fenceproblem

Maximize The Cut Segments

Longest Common Subsequence

Longest Repeated Subsequence

Longest Increasing Subsequence

Space Optimized Solution of LCS

LCS (Longest Common Subsequence) of three strings

Maximum Sum Increasing Subsequence

Count all subsequences having product less than K

Longest subsequence such that difference between adjacent is one

Maximum subsequence sum such that no three are consecutive

Egg Dropping Problem

Maximum Length Chain of Pairs

Maximum size square sub-matrix with all 1s

Maximum sum of pairs with specific difference

Min Cost PathProblem

Maximum difference of zeros and ones in binary string

Minimum number of jumps to reach end

Minimum cost to fill given weight in a bag

Minimum removals from array to make max –min <= K

Longest Common Substring

Count number of ways to reacha given score in a game

Count Balanced Binary Trees of Height h

LargestSum Contiguous Subarray [V>V>V>V IMP ]

Smallest sum contiguous subarray

Unbounded Knapsack (Repetition of items allowed)

Word Break Problem

Largest Independent Set Problem

Partition problem

Longest Palindromic Subsequence

Count All Palindromic Subsequence in a given String

Longest Palindromic Substring

Longest alternating subsequence

Weighted Job Scheduling

Coin game winner where every player has three choices

Count Derangements (Permutation such that no element appears in its original position) [ IMPORTANT ]

Maximum profit by buying and selling a share at most twice [ IMP ]

Optimal Strategy for a Game

Optimal Binary Search Tree

Palindrome PartitioningProblem

Word Wrap Problem

Mobile Numeric Keypad Problem [ IMP ]

Boolean Parenthesization Problem

Largest rectangular sub-matrix whose sum is 0

Largest area rectangular sub-matrix with equal number of 1’s and 0’s [ IMP ]

Maximum sum rectangle in a 2D matrix

Maximum profit by buying and selling a share at most k times

Find if a string is interleaved of two other strings

Maximum Length of Pair Chain

Count set bits in an integer

Find the two non-repeating elements in an array of repeating elements

Count number of bits to be flipped to convert A to B

Count total set bits in all numbers from 1 to n

Program to find whether a no is power of two

Find position of the only set bit

Copy set bits in a range

Divide two integers without using multiplication, division and mod operator

Calculate square of a number without using \*, / and pow()

Power Set

kruskal\_Algorithm.cpp

Cryptography

HeapSort.cpp

Merge\_Sort.cpp

Quick\_Sort.cpp

Randomized\_Quick\_Sort.cpp

Shellsort.cpp

Topological\_Sorting\_DFS.cpp

countsort.cpp

insertion.cpp

selection\_sort.cpp

.

PERSONAL DETAILS

Name: Sujal Maiti

Email: sujalmaiti123456@gmail.com

LinkedIn: https://www.linkedin.com/in/sujal-maiti-a4931a190

GitHub : https://github.com/sujal111

VI. Do you have any commitments during the WoC Period?

None whatsoever. My University will be closed for the entire period due to the Winter

V. Why this project

I chose NeoAlgo because I am fascinated by Data Structures and Algorithms in general. Knowing building logic, having proper grip on the DSA concepts coupled with syntax of a language coding together allows a person to be able to build whatever they can concievably think of! I am very grateful to the community for bringing around the maker movement, without it, I couldn't dream of knowing as much electronics as I do now. It will have a huge impact on the community as apparently everone learnsDSA at one point of time and they will find everything organised under one roof.

A chance to be a part of an open source commmunity: Most of my projects are based around hackathons. I have not had the opportunity to work in a large scale open source organisation, and this is an opportunity to get introduced to open source along with mentorship.

Development experience

1. HackMan (Nov 2019)

Won a hackathon in DSCE

2)Nokia Internship[June-Oct 2020]

https://docs.google.com/document/d/1cZsR39H9E4QR\_PKRqeVLCQ1VVwomOqEB/edit

https://github.com/sujal111/NOKIA.git

3)Blogs

I usually write Technical Blogs

<https://mysujalmaiti.blogspot.com>

4)Top 10 in DevFolio’s Restart India Hackathon

https://www.youtube.com/watch?v=tvf6mB2TZV4

5) Participated in HacktoberFest,HackinCodes and Autumn Open Source.

6)30 days of open source

https://drive.google.com/drive/u/0/my-drive

7) Google Cloud

https://google.qwiklabs.com/public\_profiles/e3075a35-3440-408f-b124-7d98939379bf