Learning process & Project Proposal
FOR

THE UPLIFT PROJECT

Data Structure and Algorithms

Team-03

Members Details

Mentors and Moderators:

S. no	Name	Email
1.	Shubham Tiwari (Mentor)	tiwari.98shubham@gmail.com
2.	Shatabdi Singh (Mentor)	shatabdisingh2621@gmail.com
3.	Rutuja Dhanawade (Moderator)	dhanawaderutuja1002@gmail.com

Active Participants:

S. no	Name	Email
1.	Debdyuti Das	das.debyuti01@gmail.com
2.	Apoorva Gupta	gupta.apoo98@gmail.com
3.	Aprajita Shrivastava	aprajitashrivastava100@gmail.com
4.	Bhuvenesh	vbhuvenesh05@gmail.com
5.	Charishma Boddu	charishma0249@gmail.com
6.	Arghadip Chakraborty	01argha@gmail.com
7.	Disha Sinha	disha2sinha@gmail.com
8.	Arpit Trivedi	arpittrivedi2001@gmail.com
9.	Divyanshu Rawat	be25008.18@bitmesra.ac.in
10.	Dinesh Kumar	dk4405870@gmail.com
11.	Aswin AV	aswinavofficial@gmail.com

Inactive Participants:

S. no	Name	Email
1.	Dimple Sharma	dimplesharma0798@gmail.com
2.	Divya Kelaskar	divyakelaskar@gmail.com
3.	Anushka Shukla	shuklaannushka@gmail.com
4.	Dev Arora	devro3014@gmail.com
5.	Ashis Kumar Pradhan	pradhanashish95@gmail.com
6.	Ayush Singh	ayushsingh12march@gmail.com
7.	Ayushi Panth	lit2019073@iiitl.ac.in

Learning Process

- We have divided the whole learning process into 9 weeks, and with 10th onwards we will start our project development.
- For every week we have a certain syllabus to cover, so every week mentors will provide some elite material (including tutorials) for study of those topics. (that study material is made available to participants on git-hub)
- Mentors will be conducting 2 seminars a week. In these seminars, mentors will broadly explain the topics for each week, and also solve the doubts, or queries of participants.
- Moreover, every week three assignments will be given to participants, so that they can practice what they have learnt in current and previous weeks. (All those assignments will be hosted as contests over the Hacker rank, so that progress can be tracked in a better way.)
- The whole progress of individual participants, during the The-Uplift-Project will be maintained over Git-Hub, by mentors.

Timeline:

Week - 1	**Introduction to Data Structures/ Number Theory** ● Definition of data structures and abstract data types, ● Static and Dynamic implementations, Examples and real life applications; ● Searching Algorithms: Straight Sequential Search, Binary Search, Tertiary Search. ● Number theory and Mathematical Problems, based on base conversions, prime number and sieve, Divisibility and large numbers, Catalan numbers etc.
Week - 2 and 3	**Recursion and structures/classes** ● Introduction to recursion, Divide and Conquer Algorithm. ● Euler trees formation, and system-stack memory diagrams formations for recursive functions. ● Infix, postfix, prefix representation using recursion, Conversions, Applications. ● Introduction to structures and classes. And an

	introductory level of OOPS.
Week - 4	**Queues, Lists and Stacks** ● Definition and Array based implementation of Queues / Lists. ● Linked List implementation of Queues / Lists, ● The Stacks : Definition, Array based implementation of stacks, Linked List based implementation of stacks, ● Circular implementation of Queues and Singly linked Lists, ● Straight / circular implementation of doubly linked Queues / Lists, ● Priority Queues, Applications.
Week - 5	**Running Time and Bit Manipulation** ● e, AveTime Complexity and Big-oh-notation, ● Running Times, Best case, worst casrage Case, ● Factors depends on running time, ● Evaluating Time Complexity for recursive functions. ● Bit Manipulation: Introduction to bits and binary number system, and applications of bit ● Bitwise operators and logical operators using bits. ● Tricks and tips with bits, and important tactics.
Week - 6	**Searching and Sorting** ● Introduction, Searching Algorithms: Straight Sequential Search, Binary Search ● Sorting by exchange, selection, insertions : Bubble sort, Selection sort, Efficiency of these algorithms; ● Shell sort, Performance of shell sort, ● Merge sort, Merging of sorted arrays & Algorithms; ● Quick sort Algorithm analysis, ● sortHeap: Heap Construction, Heap sort, bottom – up, Top – down Heap sort approach.
Week - 7 and 8	**Trees** ● Definition of generic trees and Binary trees, ● Properties of Binary trees and generic trees, and their Implementation, ● Tree Traversal pre-order, post order, In- order traversal, ● Binary Searching over the trees. ● Heaps and their equivalence structure with trees, ● AVL Trees, ● Implementations of the above
Week - 8 and 9	**Graphs** ● Definition of Undirected and Directed Graphs and Networks, ● The Array based implementation of graphs, ● Adjacency matrix and Adjacency list

	● The Linked List representation of graphs, ● Graph Traversal – Breadth first Traversal, Depth first Traversal, ● Shortest path Algorithm, Examples: Dijkstra, Floyd Warshall and Bellman Ford. ● Minimum spanning trees, Examples: Prims and Kruskal +algorithm. ● Tables : Definition, Hash functions, ● Implementations and Applications of the above.
Week - 10 onwards	**Project Development Phase Started:** ● Project Functioning and Strategy discussion ● Dividing the tasks, among the participants. ● Whole development would be on Git-Hub.
After the project (only if we have time)	**Dynamic Programming** ● DP: Introduction to dynamic programming, overlapping subproblems and optimal substructures. ● Approaches for dynamic programming: Memoization and Tabulation techniques.

Project Description

- Project Domain : Banking Management
- Title: Bank Records Query Optimisation (through B+ trees)

- Basic Idea:

1. This will be a DSA based project, in which we reduce the time taken to search a given record in a database, by using B-tree data structure rather than indexing and traditional sequential access.
2. Therefore, the main objective of this project is to minimise the number of disk access for certain searching of record, and achieving the techniques for arranging the data or record, can be located in as few I/O's as possible. And i should include that participants get full chances to implement and learn the application of topics that they have learnt during the learning phase.
3. The project will include an application development, which will contain lot-n-lots of bank records, (likewise bank account details and transaction details of every bank account). This application will allow bank-staff/users to initiate queries in the bank records, and those queries will be resolved within logarithmic time complexity.
4. We should mention that we are not going to use any database technologies (like SQL), the database we are talking about would be created by participants in a simple file system (totally in .txt files and properly arranged). It will make them learn more about storing data, and its retrieval.
5. The main role of Data-Structures and Algorithms in this project, will be to store the bank data records in an efficient way using B+ Trees and Linked List, creating a database in the simple file system, and designing efficient algorithms for quick retrieval of information from it.