**Implementation-of-Algorithms**

Implementation of advanced data structures and algorithms

The below are the details of the projects present

**SP0**

This has three files

* **ImplementSorts.java** - which implement the merge sort and has a priority sorted(Heap Sort) implemented.
* **MergeSortGeneric.java** - The generic version of merge sort Implementation
* **MergeSort.java** - The merge sort which only supports for integer numbers.
* This project analyzes the running time of merge sort and heap sort.
* Java's System.currentTimeMillis() has been used to analyze the running time.

**SP1**

This has three files

* **MergeSortNonRecursive.java** - The non recursive implementation of merge sort using Stacks and Queues.
* **SetOperations.java** - The has basic set operations like A-B, A∪B, A∩B with a sample driver program
* **SinglyLinkedListReverse.java** - The basic implementation of singly linked list (recursive and non recursive)

**SP2**

This has five files

The below are the graph helper classes

* **Edge.java**
* **Graph.java**
* **Vertex.java**
* **CheckEulerian.java** - To find if a given graph is Eulerian or not
* **FindOddLengthCycle.java** - This class has is to check if there is an odd length cycle in the graph for bipartite graph and also the code to check if a graph has cycle or not (both prositive and negative)

**SP3**

This has four java files

* **Fibonacci.java** - to find the nth fibonacci number.
* **MultiPivotQuickSort.java** - This has the multi pivot(two) implementation of quick sort (both random and non random)
* **QuickSort.java** - This has the single pivot implementation of quick sort (both random and non random)
* **QuickSortImplement.java** - drive to compare the running time of two versions of quick sort

**SP0PQ**

This has eight java files

The below are the graph helpers

* **Edge.java**
* **Graph.java**
* **Vertex.java**
* The below are two priority queues
* **Index.java** - helper for priority queues (PQ)
* **PQ.java** - The PQ interface
* **BinaryHeap.java** - Normal implementation of PQ's using Binary Heap
* **IndexedHeap.java** - Indexed PQ version
* The below are the MST's
* **MST.java** - The are two version of MST's here PRIM1 which uses binary heaps and PRIM2 which uses Indexed Heaps.

**SP4**

There are two files

* **BST.java** - The basic operations in binary trees like insertion, deletion, check the balance condition, check the ordering property, to find the diameter of the BST.
* **AVLTree.java** - To be modified

**LP1**

* **BigArithmetic.java** - This has basic arithmetic operations of big numbers using lists (A+B, A-B, A^b, A\*B) for base-10 numbers.

**LP2**

* This is exactly as same as the project SP0PQ except that the java file MST.java has Kruskal algorithm to find the MST