**Procedure to run the code:**

1)For bubble sort algorithm:

Array with variable name inp = [3,8,23,6,3,5,6,54,3,78,69,1,2] has been taken and bubble sort function has been applied on the ‘inp’ variable. Run the code to get bubble sort.

2)For merge sort algorithm:

Array = [1,-10,21,3,5] has been taken and merge sort algorithm has been applied. Run the code to get merge sort.

3) For growth rate algorithm:

Run the code and output will contain functions arranged in the descending order of their growth rate.

4) For binary search algorithm:

An array with variable name arr = [2,3,4,5,6,10,30,35,40,50,70,90] has been already input to the code. Enter the value to be searched in the given array. Run the code and output will contain the index of the element searched.

5) For Linked List brute force search algorithm:

Run the code. Enter the number of elements to be searched. After entering the elements, type the element to be searched.

6) For depth first and breadth first algorithm:

A graph with variable

graph= {**"a"**: [**"c"**],  
 **"b"**: [**"c"**, **"e"**],  
 **"c"**: [**"a"**, **"b"**, **"d"**, **"e"**],  
 **"d"**: [**"c"**],  
 **"e"**: [**"c"**, **"b"**],  
 **"f"**: []  
 }

has been input to the program. Run the code to get the search path using depth and breadth first algorithm.

7) For Dijkstras algorithm (Shortest Path Algorithm):

A graph containing distance has been input to the code

graph = {**"a"**: {**"c"**:5},  
 **"b"**: {**"c"**:6, **"e"**:8},  
 **"c"**: {**"a"**:12, **"b"**:5, **"d"**:2, **"e"**:1},  
 **"d"**: {**"c"**:3},  
 **"e"**: {**"c"**:24, **"b"**:5},  
 **"f"**: {}  
 }

Run the code to get the shortest distance and path between node ‘a’ and ‘e’