

Data Structures and Algorithms

Agenda

- Recursion
- Recursive Binary Search
- Sorting Algorithms
 - Selection sort
 - Bubble sort
 - Insertion sort

GIT

- <https://github.com/nilesh-g/dsa-06>
 - Java -- Classwork (Java codes)
 - day01/
 - day02/
 - Python -- Python codes -- similar to classwork
 - day01/
 - day02/
 - C++ -- C++ codes -- similar to classwork
 - day01/
 - day02/
 - Notes -- Slides + Notes + Assignments
 - day01/
 - day02/
- First time -- To copy the data
 - Open GitBash.
 - Go to directory in which data is to be copied,

- `cmd> git clone https://github.com/nilesh-g/dsa-06.git`
- For next day onwards -- To get latest data
 - Open GitBash.
 - Go to directory in which data is copied (dsa-06).
 - `cmd> git pull`
- Submit your github id into Sunbeam student portal.
 - Refer today's screenshots.

Recursion

- Why to learn recursion?
 - Problem solving techniques solved using recursion: Divide and Conquer, Backtracking.
 - Problem solving techniques depends on recursion: Dynamic programming.
 - Popular for interviews.
- Limitations
 - Need more time -- Though time complexity is same -- more time required to create FAR on stack
 - Need more space -- for FAR on stack
- Advantages
 - Programs are simpler to read/understand
- Tail-recursion
 - If recursive call is last line of the recursive function, then it is tail recursion.

Sorting Algorithms

- Sorting: Arranging elements in ascending or descending order.
- Algorithms
 - Selection sort: Select element at an index and compare with all elements after it.
 - Bubble sort: Compare two consecutive elements. Do n-1 passes.
 - Insertion sort: Find appropriate position for last unsorted element and insert it there.
 - Quick sort
 - Merge sort

- Heap sort

Assignments

1. Factorial of a given number (Using recursion).
2. Calculate power ($x ^ y$) (Using recursion).
3. nth term of Fibonacci series (Using recursion).
4. Prime factors of a given number (Using recursion).
5. Decimal to Binary conversion (Using recursion).
6. Print numbers 1 to 10 using recursion (Using recursion).
7. Implement Fibonacci search. Reference: <https://www.geeksforgeeks.org/fibonacci-search/>
8. Write a function to return number of comparisons for a bubble sort. Write another function to return number of swapping for bubble sort. Compare result for the same input array.
9. Do paperwork and calculate time complexity of insertion sort if array is already sorted.
10. Modify the insertion sort algorithm to sort the array in descending order.