Day - 3

- Space Complexity

  Space complexity refers to the total space taken by an algorithm or program including the variable data and input data sizes.

  It is total storage occupied by Variables + auxiliary space taken by program

- Time Complexity

Time complexity represents the number of times the statement is executed. It does not give actual time taken by the program but it gives a basic idea of program time irrespective of processor speeds.

  - Constant Time Complexity: O(1) : in this kind of algorithms, the time taken does not gets affected by the size of the data.
  - Linear : O(n) : the time taken increases linearly with data increase. Algorithm processes n arguments, for example , We need to print every element of an array one by one or add numbers one by one.
  - O(n log n): This is very efficient time complexity in this we use recursions to divide the number of processes into halves in every statement causing it to use only log n time to process
  - O(n2) : Nested for loops runs as n2 times as , for rvrty n loop there are another n loops.
  - Exponential (2^n): These kind of algorithms are those when we don't know the optimum method.

- Data Structures - JAVA OOB DS

  Data structures in Java includes

  1. Arrays: linear, contiguous
  2. Linked Lists: dynamic in size,
  3. Stack: It's first in last out data structure
  4. Queue: First in First out algorithm, push happens from below
  5. Binary Tree: Hierarchical structure
  6. Binary search tree: It is a sorted binary tree, Left node must be less than and right node is greater than the root node.
  7. Heaps
  8. Hash Maps: using hash functions they're mapped into address of storage.
  9. Graph: it is collections of edges and vertices. It can be directed or undirected