Tutorial 1: Basics of Algorithms
DAA
2CS503
19BCE248

Q.1 What do you mean by an algorithm? Explain the importance of an algorithm by suitable example.
An algorithm is a set of step-by-step procedures, or a set of rules to follow, for completing a specific task or solving a particular problem.Algorithmic thinking allows students to break down problems and conceptualize solutions in terms of discrete steps.

Examples such as Google Maps use graph algorithms to solve basic human problems. Thus in each and every solution pertaining algorithms plays a crucial role.

Q.2 What is the smallest value of n such that an algorithm whose running time is 100n2 runs faster than an algorithm whose running time is 2^n on the same machine?

By just substituting different values at n=15 the condition satisfies.

Q.3 Describe the method for analysing an algorithm. What do you mean by best case, average case and worst case time complexity of an algorithm?

For analysing the algorithm we analyse each and every line individually and then sum them up which gives us running time complexity and space complexity used.

Best case = fastest time to complete, with optimal inputs chosen.
Worst case = slowest time to complete, with pessimal inputs chosen.
Average case = arithmetic mean. Run the algorithm many times, using many different inputs of size *n* and then by taking the average we get an average case.

Q.4 What is the difference between the Apriori approach and Posteriori approach for solving any problem? Discuss with proper examples.

Posteriori	Apriori
It is dependent on the language of the compiler and type of hardware.	It is independent of the language of the compiler and types of hardware.
Time complexity differs from system to system	Time complexity is independent of system
Doesn't use Asymptotic notation	Uses Asymptotic notation

Q.5 Among Merge-sort, Insertion-sort and Bubble-sort, which sorting technique is the best in the worst case. Support your argument with an example and analysis.

Merge-sort gives best performance because in all three cases it's time complexity remains O(nlogn). While others have O(n^2) in the worst case.