#1

"Time Complexity & Big O notation | Analysis and Calculation | Java C++ Mukesh Bhaiya ☑️DSAOne Course #1

Mukesh Bhaiya

Time complexity is the time taken by an algorithm as a function of the length of of of input. This is time complexity in these three lines. Time complexity means an algorithm. Just as you increase its length of input then how does that algorithm perform ? How does that algorithms behave ? How do you use their constraints and tell which algorithm is to be used here ? Time complexity has been invented and it basically tells us that which is the better algorithm among the two ? And there comes in use - The Big O Notation. This Big O notation works to quantify the time complexity of an algorithm. It tells that the this algorithm will take this much time according to this length of input. Big O notation takes in account the bigger power and nothing other than that. For any n which one is greater, obviously n is greater. Even if the constant is large we 'll assume it as O (1) Even if there is a very large no. with n^2 then we 'will still consider it O (n^2) If you will make the graph of n will be something like this.

Big O notation is mostly used in computer science because we remain safe by finding through Big O notations for multiple algorithms. We consider the META notation. OMEGA NOTATION tells that what will be the lower bound of the algorithm. THETA and OMEGA notation are other notations that tell us about the time complexity of algorithms. Using the tree method, we 'll try to find with the. tree method so that you can easily find the time complexity of recursive functions. First of all, you write the non-recursive part in this way- n c. after that the. recursive part is T (n/ 2) and the. recursion part. After adding them then you 'll have n c operations performed. You have to see the no. of operations performed at each step. If you want to tell which algorithm is better according to the constraint in the online judges. Then what is the method. I have already told you about this in a previous video. But if you still want to know then I 'll tell you. Try finding their time complexities and tell me what did you find. I'll put their answers in my telegram channel.

You just need to check that 10^8 operations rule is being followed and you can easily find if time constraint is given to me then what will be the algorithm. Similarly if its given between 15-18 then this type of complexity will be accepted. If its given 100 then you can go till (n^4) If 400 is given, the you can then go till n^3. If you found that its - n log n , then the algorithms based on it , you should strike at that time. It means I cab either apply sorting- it might be solved after sorting or such an operation that can be don in - log n I can perform that operation n times , you can"