TIME COMPLEXITY

Hi guys,

Hope you have started coding :)

In this document, we will learn about time-complexity.

Normally we want efficient algorithms and we want our computations even faster.
The most interesting ways we classify problems is by asking questions about how time-efficient algorithms are for problems.Computation has no benefit if it takes exceedingly long to solve a problem.So we will learn here about how to calculate time-complexity of a particular function or code.The time complexity of an algorithm is the total amount of time required by an algorithm to complete its execution.This time required is basically number of operations required while executing a particular code.

In simple words, every piece of code we write, takes time to execute. The time taken by any piece of code to run is known as the time complexity of that code. The lesser the time complexity, the faster the execution.

Tutorials:

https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/

https://m.youtube.com/playlist?list=PL2_aWCzGMAwI9HK8YPVBjElbLbI3ufctn
(Refer only last video of the above link)

https://www.geeksforgeeks.org/analysis-of-algorithms-set-4-analysis-of-loops/

By learning time complexity, sometimes while doing a code we can get idea that this code can be run in $O(n^2)$, $O(n^3)$ or whatever by looking at the constraints. So we can look forward a solution in which the solution does not exceed the given bound.

Questions:

Try to solve all of these for practise purpose.

https://www.geeksforgeeks.org/practice-questions-time-complexity-analysis/

https://www.geeksforgeeks.org/interesting-time-complexity-question/

https://www.geeksforgeeks.org/a-time-complexity-question/

https://www.interviewbit.com/problems/nestedcmpl3/

https://www.interviewbit.com/problems/loopcmpl2/

https://www.interviewbit.com/problems/gcdcmpl/

https://www.interviewbit.com/problems/amortized1/

Extra:

If you don't get confidence, try these until you become confident enough.

https://www.geeksforgeeks.org/algorithms-gq/analysis-of-algorithms-gq/