Overview Q&A Downloads Announcements

# Introduction to Data Structures & Algorithms

Greetings everyone, and thank you for joining me on this new playlist of mine, Data Structures, and Algorithms. You'll learn everything needed to prepare for interviews, placements, and logic building in this course. This playlist will teach you how to create optimal solutions to your real-world problems. This course explores a broad range of data structures and algorithms. Furthermore, I have designed this course with the optimal requirements in mind and will try my best not to miss anything.
The only thing you need to know is the fundamentals of C/C++ to get started with this course. If you are not yet ready with that, you may want to access my short tutorials on C and C++. For now, that will suffice. Links are here:

- **C**: C Programming Tutorial For Beginners: Learn C In Hindi
- **C++**: C++ Tutorial For Beginners: Learn C Plus Plus In Hindi

Although you can start here, even if you know Java well. The reason for adhering to C or C++ is to be able to write the code from scratch. These are very recommended since they make it easy to understand the fundamentals of these theories.
Note: Do not forget to access the notes I have provided along with the course. So, without any further ado, let's start our Data structures and Algorithms journey.

## Data Structures and Algorithms:

Let's clear up our basics with these terms before deep diving into DSA. Data Structures and Algorithms are two different things.
**Data Structures** – These are like the ingredients you need to build efficient algorithms. These are the ways to arrange data so that they (data items) can be used efficiently in the main memory. Examples: Array, Stack, Linked List, and many more. You don't need to worry about these names. These topics will be covered in detail in the upcoming tutorials.
**Algorithms** – Sequence of steps performed on the data using efficient data structures to solve a given problem, be it a basic or real-life-based one. Examples include: sorting an array.

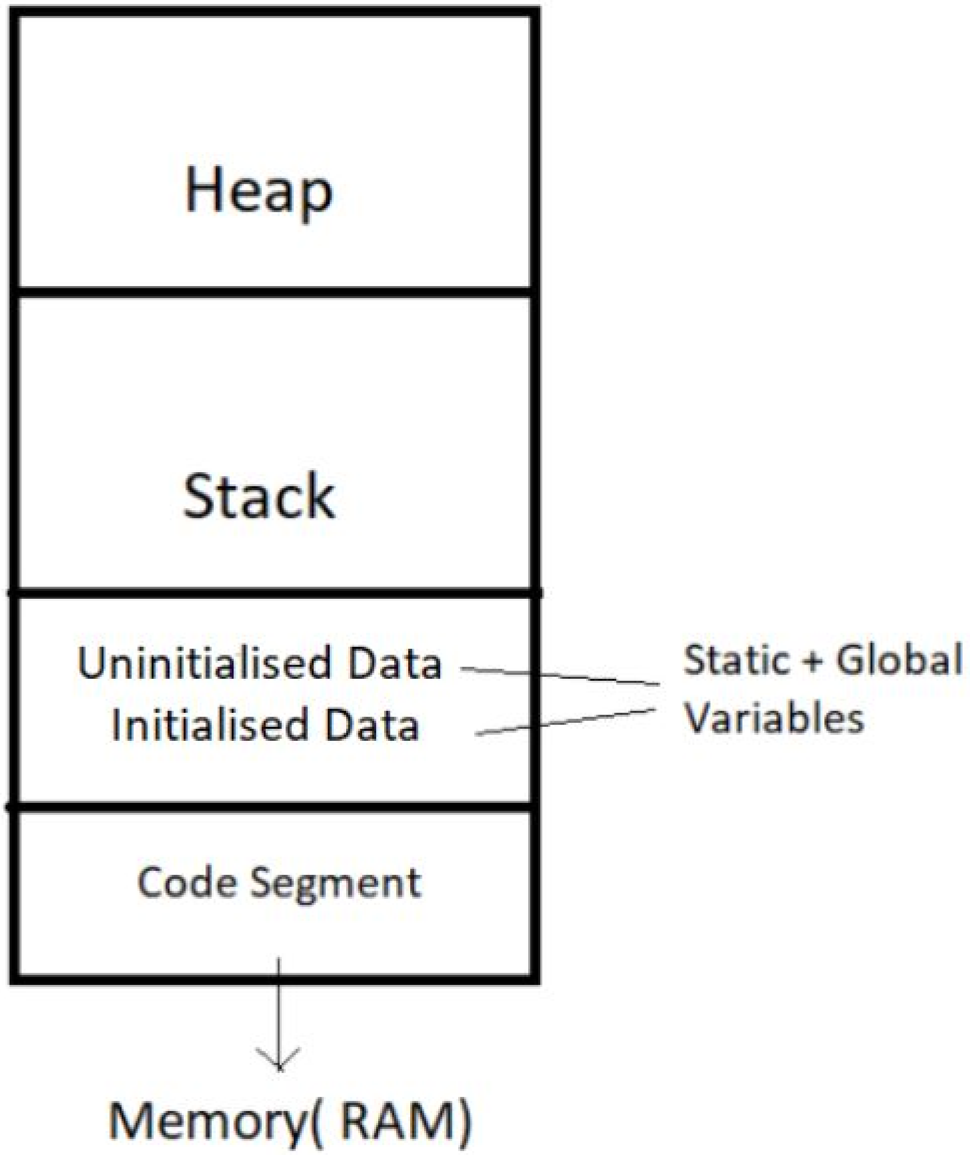
## Some other Important terminologies:

1. **Database** – Collection of information in permanent storage for faster retrieval and updation. Examples are MySql, MongoDB, etc.
2. **Data warehouse** – Management of huge data of legacy data( the data we keep at a different place from our fresh data in the database to make the process of retrieval and updation fast) for better analysis.
3. **Big data** – Analysis of too large or complex data, which cannot be dealt with the traditional data processing applications.

## Memory Layout of C Programs:

- When the program starts, its code gets copied to the main memory.
- **The stack** holds the memory occupied by functions. It stores the activation records of the functions used in the program. And erases them as they get executed.
- **The heap** contains the data which is requested by the program as dynamic memory using pointers.
- **Initialized and uninitialized data** segments hold initialized and uninitialized global variables, respectively.

Take a look at the below diagram for a better understanding:



So, this was all for the beginning. Data Structures and Algorithms are not new concepts. If you have done programming in any language like C, you must have come across  Arrays – A data structure. And algorithms are just sequences of processing steps to solve a problem. :)

Thank you for being with me throughout the tutorial. I hope you enjoyed it. If you appreciate my work, please let your friends know about this course too. Make sure to download the notes and make the most of this course! See you all in the next tutorial, where we'll learn about Time Complexities. Till then, keep learning and keep coding.