

Summer of Science DSA

For those who don't know the timeline.

The timeline for SoS 2023 is as follows:

21st May - Plan of action submission: This short document serves as a roadmap for your summer goals. It should outline the topics you plan to cover, provide a week-by-week timeline for your progress, and detail your approach to tackling these subjects. Your mentor will assist you in crafting the Plan of Action. So please ensure you have established proper contact with your mentor and have had meaningful interactions. If your mentor isn't responding please contact us as soon as possible.

20th June - The Mid-summer report submission: The report will have two parts. The first part will focus on the progress you have made until June 18th. The second part will involve presenting your modified plan of action, which may differ from your original plan. Reflecting on your progress and adjusting your plan accordingly will allow you to better guide your efforts and work towards achieving your ultimate goal.

20th July - SoS report submission and video submission: The submission of the final report and video will mark the completion of the SoS project. The mentor will verify and validate the project before considering it as complete. Certificates will be issued solely to those participants who successfully finish their SoS projects. Upon your consent, the completed reports will be uploaded on the MnP website, enabling the science community within the institute to benefit and learn from them. Further information regarding the video will be provided at a later stage..

To ensure the regularity of the process, we will not be considering projects for further stages if we do not receive your Plan of Action or Midterm Report. Any requests regarding the same will not be entertained.

Resources

For People who are new to this topic

Playlist —

- [Lectures](#) (leave lectures from 11 - 16)
- [Slides for the above playlist](#)

Readings —

- [DSA Quick Revision](#)
- [book](#)

For Freshies —

Keeping in mind about the ongoing semester and also as you have plenty of time to prepare and practice even in further years you can go at a slow pace too which is fine.

Task 1 Compulsory —

You can have a look at 2-3 lectures a week and implement the data structure/algorithm discussed in that lecture. Reference implementation of the data structure/algorithm discussed can be found in either of the links provided in the readings section. I recommend you to read the book provided as a link in the readings section for reference implementation of the data structures. Reference implementation can be done in your favourite programming language.

Task 2 Optional —

After implementing the data structure/algorithm discussed in the lecture you can even do 2-3 problems related to that data structure/algorithm, questions related to a specific data structure can be found in the first link of the Readings section. But please **note this** to solve the problems you need to know about [STL Library](#) / [4th Chapter in this book](#) for C++, [Collections](#) for Java, [Collections](#) for python. So before solving the problems have a look about the data structure you are using in that week.

Note —

You can follow the lectures according to the order given in the playlist and note that Task 1 is compulsory and Task 2 is optional as it is an ongoing semester please do not prioritise this SoS than your academic commitments as **CPI** is important than this during first year 😊, this dsa can even be done in the second year too but first year cpi can not be gained again (I am saying according to the old curriculum as most of the courses are MPC sorry if the curriculum has been changed). Even though the new curriculum is different I strongly believe that first year is the time where we can gain a lot of cpi (personal opinion 😊).

For People who are new to this topic

For Sophies and Third year people —

Keeping in mind about the incoming internship season or placement season, people who want to pursue their career in the software side, you need to have a great understanding about data structures and algorithms. So this is for all those who want to get into a software profile. As you are new to this topic I recommend you to give a significant amount of time every day to this.

Following are the tasks that need to be done

Task 1 —

So the plan is to try to have a look at about 2-3 lectures per week and implement data structure/algorithm discussed in that lecture. Reference implementation of the data structure/algorithm discussed can be found in either of the links provided in the readings section. If you are not having any other commitments then I recommend to have a look at about 4-5 lectures per week and also read the book provided as a link in the readings section for the reference implementation of the data structures, but if so you are having some commitments like an (intern or any research projects etc) you can have a quick look at the implementations provided in the first link of the reading section and have a look at about 2-3 lectures per week.

Task 2 —

After implementing the data structure/algorithm discussed in the lecture I recommend doing 2-3 problems related to that data structure/algorithm, questions related to a specific data structure can be found in the first link of the Readings section. But please **note this** to solve the problems you need to know about the [STL Library](#) / [4th chapter in this book](#) for C++, [Collections](#) for Java, [Collections](#) for Python.

So before solving the problems have a look about the data structure you are using in that week.

Both Task 1 and Task 2 are highly recommended if you are preparing for a software role in the incoming intern season or placement season. Once this is done now you can practise questions. For practising you can follow the websites provided below.

For people who already have an idea on DSA

As you already have an idea regarding the topic instead of looking at lectures or playlists I would highly recommend you to practise as many problems as you can, following are some of the good websites to practise(personal selection).

First have a brief look at the data structures in the following link and also first 4 chapters from the hand book provided and then start practising.

- [Quick revision](#)
- [handbook](#)

As you require questions category wise, following links will help you to practise questions sorted in a proper way. You can choose any of the first three, the last link contains few questions compared to the first three.

- [NeetCode](#)
- [CSES](#)
- [Striver's Sheet](#)
- [WnCC Practise Questions](#)

If you have any doubt regarding any algorithm you can have a look in the following two links

- [Algorithms](#)
- [Handbook](#)

Please tell me if you have already practised in the above provided links I will provide some more, If you haven't done problems present in the above links then believe me those links contain a good collection of problems.

If you are preparing for interviews and want to have a quick overview of OOP you can have a look at this [Interviewbit](#).

If you are a beginner for OOP then this links will provide a good insight

- [OOP](#) for C++ [Playlist](#) starts from Lecture 33
- [OOP](#) for Java [Playlist](#)
- [OOP](#) for Python [Playlist](#)

!!! NOTE !

The topic that you have chosen is more of a practice oriented one, of course the theoretical part is also significant. So keeping this in my mind I will not care about the report which you write, I will just see the amount of practice you have done for this topic. So if possible please maintain a Github repo which keeps track of each week's progress by placing all the code files which you have written in that week with proper naming to the file for example if you have implemented a stack then name of the file can be stack.*(* can be anything depending on your programming language).

To get a certificate you need to write a midterm, a full term report and a demo at the end, so for name sake write a report, you can write the report using [this](#). I don't even care even though you copy and paste the same, but the thing which I will check for sure is the amount of practice you do according to the plan of action you provide and according to your weekly progress. It's fine even though the progress is half of the syllabus provided to you or half the syllabus according to the plan of action provided by you.

If you are having a good amount of time and interested in making a good report, that would be a great thing because you can use that report as a hand book which helps you in revising things quicker just before an interview/exam. And do place that report in your repo.

So first prepare a plan of action according to the syllabus provided before the deadline provided in the mail and send that to me before submitting that plan of action, I will suggest some changes if there are any.

Also before practising if you are not able to recollect the basics of that programming language which you are using feel free to have a week in your plan of action for revising those basics probably the first week.

- If you are having the same experience in using Java or C++ and if you are not sure which language is best to use I would recommend you to use C++ as using C++ is fun 😊.
- If you have a lot of experience in one language please follow that language. Above said suggestion is a personal opinion.

By the way sorry for the video quality of the lectures provided, but they are good according to the explanation of theory and content.

HAPPY LEARNING