**These are some of the compiled resources that you may want to look at along with all the topics listed in the CSE221 Mid term syllabus**

1. **Iterative Time Complexity:** You may have to extract the instructions from the code [See the below doc links and previous semester questions for reference]. Make sure you understand how to analyse the code and get the time complexity, You need to know the series formulas that are mentioned in my doc files **[GP series: Most dependant nested loop follows this series, summation series, root series, log series(done in the recursive time complexity note) etc.]** All maths with the series formula can be found in those working sheets I provided:

[1. Algorithm introduction + Time complexity [HFN] Part1](https://docs.google.com/document/d/1pdqqpPAqyf1XBsIsYOX1FDKoPk9tYmzpSZ-66eqlRjI/edit?usp=drive_link)

[1.3 time complexity Additional Solution and prep.pdf](https://drive.google.com/file/d/1ds5EIQ3jK6DXIdHOXBJ3HVJ4tUWaowQz/view?usp=drive_link)

G.P. series formula given on this file:

[2.2 Recursion time complexity HFN additional.pdf](https://drive.google.com/file/d/1vB5MxvMSXM96t6p0oZ2GMWPwAMPgibCI/view?usp=drive_link)

Some additional YouTube resources for basic T.C. calculation in case you forgot:

<https://youtube.com/playlist?list=PLVqu3pu23LttpphV1fwFIlxHQd3THX7Wj&si=DWEPOhFJhOIdaBUb>

1. **Recursive Time Complexity:** There are three methods, **Recursion Tree, Substitute Method and Master’s Theorem.** Keep in mind that for master’s theorem, you need to show the format of the equation and all the conditions it needs to satisfy instead of only writing the answer directly. You may need to extract the time series from the code or you will be given the time series as a function for you to solve. Both are written and demonstrated in the notes and classes.

[2. Recursion time complexity HFN.pdf](https://drive.google.com/file/d/1yD1e8Eml1AWESkaLi_Ov0jYG75g0F2Hm/view?usp=drive_link)

[2.1 Time complexity Part2 [recursion] [supplements]](https://docs.google.com/document/d/1seXSDmvCcmuASihXh8T2JGqAhYVD_KMJD3VwDViwdII/edit?usp=drive_link) [There are tutorial videos in this doc if you missed the class]

[2.2 Recursion time complexity HFN additional.pdf](https://drive.google.com/file/d/1vB5MxvMSXM96t6p0oZ2GMWPwAMPgibCI/view?usp=drive_link)

1. **Searching:** You need to understand how binary search, linear search, ternary etc work and how they differ from each other. Also, pseudocode/code/implementational logic is needed for searching [check previous semester question pattern]. I have shown you the difference between binary vs ternary in the class. And also, use the idea of those search techniques to achieve something as the task suggests. Knowing the time complexity and use case of each search is a must for you to implement it [use previous semester questions as reference].

[3. Search [linear, binary, ternary]](https://docs.google.com/document/d/1kIId7CmEYATxLu4Uvn9koTBbLYCOFvBMRWSY3KMHZTY/edit?usp=drive_link)

Some additional YouTube resources for many implementations of Binary Search:

<https://youtube.com/playlist?list=PL2_aWCzGMAwL3ldWlrii6YeLszojgH77j&si=MYFP0Z8XcASxZAoI>

1. **Divide and Conquer:** We learned **Merge Sort, Quick Sort, Karatsuba, and Max sum Subarray**. You have to learn the process/ pseudocode/step-by-step logic/instructions[check previous semester question pattern]. For sorting, simulation and knowing how each of these two sorting algorithms works are equally important. You will not be directly asked to solve a problem you have to add some logic to solve it like your Quiz 02 question.

[4. Divide & Conquer [HFN]](https://docs.google.com/document/d/1vvqJs1N7yTVeUDqCXIdkoso1f-kz-jhddOOIEYJ2RpI/edit?usp=drive_link) [Only the topic I mentioned]

[5. divide and conquer HFN [closest pair, max sum subarray, karatsuba].pdf](https://drive.google.com/file/d/13uXoXn1Vy8A8RHEiZrt_h3OLz3dUOP3_/view?usp=drive_link)[Only the topics I mentioned]

1. **Graph:** The topics you must cover for Graph are: The definition and properties of a graph, representation of a graph [adj matrix & adj list], Traversal of the graph [BFS, DFS] and implementation of the graph traversal algorithm such as for BFS: Shortest path calculation, Bipartite graph detection. For DFS: Edge Classification and Cycle Detection. I have not covered Cycle detection properly in the class since it was a part of the lab as well. However, **I am providing the recordings where everything except the bipartite graph is covered for the whole graph syllabus**

1. **Lecture 1:** **Intro to graph + BFS + shortest path:** <https://youtu.be/-kP2g7IfZr4?si=WI9V4KBU9nPec68x> **[bipartite missing here]**
2. **Lecture 2:** **DFS + edge classification + Cycle detection:**

<https://youtu.be/LDz1sZSL0Y8?si=_HlXEjJZA1N5VNgX>

1. Supplement Video to identify the **Bipartite Graph:** [How to Tell if Graph is Bipartite (by hand) | Graph Theory](https://www.youtube.com/watch?v=bZBmN7I7GNQ)
2. This semester’s class recording, where I showed edge classification and DFS [not needed if you watch the previous 2 recordings I attached]

<https://youtu.be/O1mII8stMZc?si=jsy9OUGhtpaSn5sL>

**Practice Problems From:**

* **Theory Assignment 1:** [[MARKED] SUMMER 24\_Theory\_Assignment\_1\_HFN.pdf](https://drive.google.com/file/d/1B3e-szSCTEhye-pR4cgf9W7S5DdoP8O0/view?usp=drive_link)
* **Additional Graph Assignment from the previous semester:**

[[practice] Graphs.pdf](https://drive.google.com/file/d/1L-fOjpa0D70935mi2o9HQYuswDz3jDKl/view?usp=drive_link)

* Quiz 1 & Quiz 2 Question: [Quiz Questions](https://drive.google.com/drive/folders/13sFLjE5PJO2SGL_l6RNF6y83t08XX-Ar?usp=drive_link)

**BEST OF LUCK!!**