Recursion is a case where we solve the problem by dividing the problem into sub-problems and keep repeating that until we hit a base-case and then returning and accumulating the solution as it returns back to the original caller. [Please practice recursion from CSE220 if you forgot. Solving some basic recursion problem will suffice also.

**The handwritten note for this time complexity analysis for different recursive scenarios are given in this pdf:**

[**Recursion time complexity analysis**](https://drive.google.com/file/d/142p2OcEJS_h_VbYo6ra3t6fUYSdBRTwz/view?usp=sharing)

[OPTIONAL FOR FURTHER STUDY IF YOU REALLY WANT]

The 3rd approach named “Master Theorem” is formulated and the mathematical derivation of that theorem can be found here:

[**Recursive time complexity note Stanford**](https://stanford-cs161.github.io/winter2021/assets/files/lecture3-notes.pdf)

You can see these lecture classes from MIT if you really want to:

[**MIT Algo Lecture 2 Fall 2005**](https://www.youtube.com/watch?v=whjt_N9uYFI&list=PL8B24C31197EC371C&index=2)

[**MIT Algo Lecture 3 Fall 2005**](https://www.youtube.com/watch?v=-EQTVuAhSFY&list=PL8B24C31197EC371C&index=3)

**Also, These may be very complicated and redundant for some people. You can also watch this playlist [Video 2 to 2.5]**

[**Abdullah Bari lecture series**](https://www.youtube.com/playlist?list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O)