|  |  |
| --- | --- |
| **Unit** | **Description** |
| **1** | **Analysis of Algorithm Based on Time:**   * 1. **Asymptotic notations:** Omega, Theta, Big-O, Small-o, small Omega and Tilde   ([Analysis of algorithms | little o and little omega notations - GeeksforGeeks](https://www.geeksforgeeks.org/analysis-of-algorithems-little-o-and-little-omega-notations/))\  ([Big-O, Little-o, Theta, Omega · Data Structures and Algorithms (gitbooks.io)](https://cathyatseneca.gitbooks.io/data-structures-and-algorithms/content/analysis/notations.html))  ([Difference between Big O vs Big Theta Θ vs Big Omega Ω Notations - GeeksforGeeks](https://www.geeksforgeeks.org/difference-between-big-oh-big-omega-and-big-theta/))  ([Big Oh(O) vs Big Omega(Ω) vs Big Theta(θ) notations | Asymptotic Analysis of Algorithms with Example (youtube.com)](https://www.youtube.com/watch?v=1tfdr1Iv6JA))  [Asymptotic notation | Big omega, little oh & omega | Design & Algorithms | Lec-10 | Bhanu Priya (youtube.com)](https://www.youtube.com/watch?v=vLl1BAotQ60)   * 1. **Amortized Analysis:** Aggregate Method, Accounting Method, Potential Method   (refer class notes and videos shared at the time of term examinations)   * 1. Beyond worst-case analysis   (refer notes attached with this email)   * 1. Dynamic tables and its amortized analysis   ([Lecture 21: Amortized analysis and dynamic tables (cornell.edu)](https://www.cs.cornell.edu/courses/cs3110/2009sp/lectures/lec21.html))   * 1. RAM model analysis of algorithm   [RAM Model of Computation | Algorithm analysis | Time Complexity of Algorithms (youtube.com)](https://www.youtube.com/watch?v=xbVRHTGor70)  [What is the RAM model, and what are the best, worst, and average-case complexity of algorithms? | by Explore Intellect | Medium](https://medium.com/@exploreintellect/what-is-the-ram-model-of-computation-a5e4a7ce22b4) |
| **2** | **Probabilistic and Randomized Algorithm:**  **2.1** Probabilistic approach to algorithm and Randomized Analysis  (refer class notes and coreman book chapter 5)  **2.2** Indicator Random Variable (IRV)  (refer class notes and coreman book chapter 5)  **2.3** Randomized Quick Sort  (refer class notes, also attached with this email)  **2.4** Analysis of Hiring Problem (Flipped Classroom: Analysis of Birthday Paradox Problem, Bins and Balls Problem using IRV)  (refer class notes and coreman book, also refer notes attached with this email)  **2.5** Numerical Probabilistic algorithms with example (refer notes attached)  **2.6** Las Vegas and Monte Carlo algorithm  ([Randomized Algorithms | Set 2 (Classification and Applications) - GeeksforGeeks](https://www.geeksforgeeks.org/randomized-algorithms-set-2-classification-and-applications/))  **2.7** Game theoretic randomized algorithm techniques (Tic-Tac-Toe)  ([Finding optimal move in Tic-Tac-Toe using Minimax Algorithm in Game Theory - GeeksforGeeks](https://www.geeksforgeeks.org/finding-optimal-move-in-tic-tac-toe-using-minimax-algorithm-in-game-theory/))  Also see notes attached for game theory |
| **3** | **Advanced Data Structures:**  Red-Black Tree (refer class notes)  , Randomized BST (refer class notes),  Tango Tree ([Tango Tree Data Structure - GeeksforGeeks](https://www.geeksforgeeks.org/tango-tree-data-structure/))  [Tango Tree Data Structure - javatpoint](https://www.javatpoint.com/tango-tree-data-structure)  Binomial Tree (refer class notes)  Binomial Heap (refer class notes)  Treap ([Treap - Algorithms for Competitive Programming (cp-algorithms.com)](https://cp-algorithms.com/data_structures/treap.html))  KD Tree (refer class notes)  R Tree (refer class notes)  ([Introduction to R-tree - GeeksforGeeks](https://www.geeksforgeeks.org/introduction-to-r-tree/))  [Basics of R Tree (opengenus.org)](https://iq.opengenus.org/r-tree/)  [The R-Tree: A dynamic index structure for spatial searching (hpi.de)](https://hpi.de/rabl/teaching/winter-term-2019-20/foundations-of-database-systems/the-r-tree-a-dynamic-index-structure-for-spatial-searching.html)  (simple video: [R-Tree (youtube.com)](https://www.youtube.com/watch?v=Jd8F2hVnGtQ))  [R-Trees | Data Structures and Algorithms for Big Data (youtube.com)](https://www.youtube.com/watch?v=42_gAAkEf3k)  LogLog and HyperLogLog  [HyperLogLog: How to estimate cardinality in extremely large datasets using little memory and time? | by Cheng-Wei Hu | 胡程維 | Medium | Towards Data Science](https://towardsdatascience.com/hyperloglog-a-simple-but-powerful-algorithm-for-data-scientists-aed50fe47869)  [The Algorithm with the Best Name - HyperLogLog Explained #SoME1 (youtube.com)](https://www.youtube.com/watch?v=2PlrMCiUN_s)  [Hyperloglog: Facebook's algorithm to count distinct elements (youtube.com)](https://www.youtube.com/watch?v=eV1haPUt0NU)  Count Min sketch  [Advanced Data Structures: Count-Min Sketches (youtube.com)](https://www.youtube.com/watch?v=mPxslXpg8wA)  [Count Min Sketch (youtube.com)](https://www.youtube.com/watch?v=lGoCslwItiU)  MinHash with Data mining context  [Min Hashing and Jaccard Similarity (youtube.com)](https://www.youtube.com/watch?v=OENSdmiRMOM)  Quadtree  ([Multidimensional Data, Video 4, Quadtrees and Quadtree Insertion (youtube.com)](https://www.youtube.com/watch?v=iG1o0qOyZlA))  [Quad Tree - GeeksforGeeks](https://www.geeksforgeeks.org/quad-tree/) |
| **4** | **Graph Based Algorithms:**  **4.1 Flow Network Introduction:** Residual Network, Augmenting Path, Ford-Fulkerson Method, Edmonds-Karp Method, Push-Relable Algorithm  **4.2 Bipartite Matching:** Maximum Bipartite Matching, Weighted Bipartite Matching, Weighted Non-Bipartite Matching (Edmonds algorithm)  **4.3 Max Flow Min Cut**  (Refer class notes and shared ppts) |
| **5** | Line Segment Properties (refer coreman book)  [Computational Geometry: Line Segment Properties ( Two lines Clockwise or Counterclockwise) (youtube.com)](https://www.youtube.com/watch?v=3YFUQDRL1s4)  [Line-segment Properties (skedsoft.com)](https://www.skedsoft.com/books/design-analysis-of-algorithm/line-segment-properties)  Convex Hull Graham’s scan algorithm (refer class notes)  Point Location in polygon using Ray Crossing  [Point in polygon - Wikipedia](https://en.wikipedia.org/wiki/Point_in_polygon)  [Ray-casting algorithm - Rosetta Code](https://rosettacode.org/wiki/Ray-casting_algorithm)  [Raycasting Algorithms Part 1 (youtube.com)](https://www.youtube.com/watch?v=ebzlMOw79Yw)  [Checking if a point is inside a polygon is RIDICULOUSLY simple (Ray casting algorithm) - Inside code (youtube.com)](https://www.youtube.com/watch?v=RSXM9bgqxJM)  River Search Problem, Competitive Ratio, K-Server  (refer notes attached)  [Find the minimum cost to cross the River - GeeksforGeeks](https://www.geeksforgeeks.org/find-the-minimum-cost-to-cross-the-river/)  (simple notes: [Course Materials (mit.edu)](https://6.5210.csail.mit.edu/materials.html))  [6.854 -- Advanced Algorithms (mit.edu)](https://courses.csail.mit.edu/6.854/05/Handouts/info-html/)  [Work Function Algorithm to solve k-server problem (part 1) (youtube.com)](https://www.youtube.com/watch?v=oxd9JfoU_lU)  [Work Function Algorithm to solve k-server problem (part 2) (youtube.com)](https://www.youtube.com/watch?v=tbQfRbfoyJw) |
| **6** | P, NP, NP Hardness and NP Completeness (refer shared ppt)  [8. NP-Hard and NP-Complete Problems (youtube.com)](https://www.youtube.com/watch?v=e2cF8a5aAhE)  Satisfiability(3 sat), Reducibility, TSP (Flipped Classroom: Sum of Subsets)  [Travelling Salesman Problem is NP complete (youtube.com)](https://www.youtube.com/watch?v=IW5EfD84clE)  Vertex Cover Problem, Travelling Sales Person problem (refer coreman book chapter 35 for approximation algorithms, really simple explanation on one page each)  Randomized Rounding, Primal Dual algorithms  RP, BPP, ZPP (Adleman’s theorem)  [mod06lec31 - Comparison Between Randomized Complexity Classes (youtube.com)](https://www.youtube.com/watch?v=k2nzRrQNoTg)  [The Complexity Class RP (youtube.com)](https://www.youtube.com/watch?v=5wV_FXUA3w8)  [The Complexity Class BPP (youtube.com)](https://www.youtube.com/watch?v=CinKHIkJKu0)  Also see notes attached  Turing Machine Halting Problem  [The Halting Problem (youtube.com)](https://www.youtube.com/watch?v=6XZvw9W9QSc)  [Undecidability of the Halting Problem (youtube.com)](https://www.youtube.com/watch?v=_eM0-KfAmhQ)  <https://www.geeksforgeeks.org/types-of-complexity-classes-p-np-conp-np-hard-and-np-complete/>  [7.2: The Halting Problem - Engineering LibreTexts](https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Mathematics_for_Computer_Science_(Lehman_Leighton_and_Meyer)/01%3A_Proofs/07%3A_Infinite_Sets/7.02%3A_The_Halting_Problem)  [halting problem | Turing Machine(TM) | TOC | Lec-95 | Bhanu Priya (youtube.com)](https://www.youtube.com/watch?v=0Q9qAM2htII) |

Coremen book chapters to refer:

1. Module 1

1.1 complexity : chapter 3 and 4

1.2 amortized analysis: chapter 17

      2. Module 2 : chapter 5

      3. Module 3: Red Black Tree, chapter 13

      4. Module 4: flow Network and ford Fulkerson, chapter chapter 26

Videos to refer:

1. amortized analysis using dynamic tables : <https://www.youtube.com/watch?v=iy-WhloN6vA>
2. Probabilistic analysis Hiring Problem: <https://www.youtube.com/watch?v=BD-NJekPgsY>
3. Probabilistic analysis Randomized algorithms: <https://www.youtube.com/watch?v=u1z-1QVdm9I>
4. Indicator Random Variable(IRV): <https://www.youtube.com/watch?v=xVQm3eTbmqs>
5. Hiring Problem using IRV: <https://www.youtube.com/watch?v=yQAw564S-Xg>
6. Birthday Paradox: <https://www.youtube.com/watch?v=1tnas6FQxX8>
7. Balls and Bins: <https://www.youtube.com/watch?v=OOYl7_D2LvU>
8. Red Black Tree Violations: <https://www.youtube.com/watch?v=axa2g5oOzCE>
9. Red Black tree Rotation: <https://www.youtube.com/watch?v=PhY56LpCtP4>
10. Red Black Tree Insertion: <https://www.youtube.com/watch?v=UaLIHuR1t8Q>
11. Red Black tree Deletion: <https://www.youtube.com/watch?v=CTvfzU_uNKE>
12. Red Black tree Deletion: <https://www.geeksforgeeks.org/red-black-tree-set-3-delete-2/>
13. Red Black tree Deletion: <https://www.cs.purdue.edu/homes/ayg/CS251/slides/chap13c.pdf>
14. Ford Fulkerson: <https://www.youtube.com/watch?v=GiN3jRdgxU4>
15. Master Method examples with solution: <https://www.youtube.com/watch?v=lPUhHmgrpik>

**With all this also refer the notes which were shared during respective lectures.**