Karatsuba algorithm:

1. <https://youtu.be/7_opFwubodM?si=9oyAMZroFTRspUOY> (Good)

Recursive Tree method:

1. <https://youtu.be/Eqn1cjl09e4?si=_O4X1OQ236dMSiq3> And <https://youtu.be/tl87EJby2cs?si=i0ClxBVuRCQBUpNX>
2. <https://youtu.be/vmqe-Wo7ghU?si=0bCigoXgyI2GttrQ>

Master Theorem:

1. <https://youtu.be/FBKjvXGGCJM?si=o-FzeqlS-O8CwZ3G> and <https://youtu.be/nNabmfua14c?si=-DJ7hYQc-kGyXg3r> and <https://youtu.be/gazsHlpcfCE?si=-1Q-JTlHpv28GyI8> (BEST, BEST, BEST, SUPER, SUPER, SUPER)

Graph:

1. Basic Terminologies:
   1. Directed graph (Self edge) and Undirected graph
   2. Weighted graph and Unweighted graph
   3. Degree of vertices
      1. Indegree
      2. Outdegree
      3. See example for Directed graph and Undirected graph
      4. Theorem:
         1. sum(deg(V)) = 2\*edges; Undirected graph
         2. sum(deg(V)) = edges; Directed graph
         3. sum(indeg(V)) = sum(outdeg(V)) = edges
   4. Path
      1. Simple Path
      2. Cyclic Path
   5. Connected graph and Unconnected graph
      1. Strongly Connected Graph
   6. Tree
2. Graph Representation
   1. Adjacency Matrix and Adjacency List
      1. Need to know what it is.
      2. Advantage, Disadvantage
3. Graph Traversal
   * 1. BFS-DFS both:
        1. <https://youtu.be/N2P7w22tN9c?si=-xkZ1kUlK_GO0k6K> (Chole)
   1. **BFS**:
      * 1. <https://youtu.be/GVSxGBCtQCE?si=duZmobszKccqhJvo> (ekdom Basic)
        2. <https://youtu.be/-Ja252yEllE?si=FAiM2o7x_qndMXpi> (Chole)
        3. <https://youtu.be/xlVX7dXLS64?si=OTmErxb1jb3aHL5B> (Best, Best, Best for learning what DFS is, simulation and pseudo code; Need for the Assignment)
      1. Simulation (Distance Related)
         1. <https://youtu.be/bgLecRSOWNM?si=189HgRHwrO7C1HwW> (Good, taught in the class)
   2. **DFS**
      * 1. <https://youtu.be/Lax2okU8yM8?si=24tZeVa7Kf-ZUg8o> and <https://youtu.be/CySyAj1v1rA?si=_hD0PfCpc73HA3fO> (Motamoti)
        2. <https://youtu.be/I7aVemydDTU?si=wICBxh0xmvcPP8OT> (Chole)
        3. <https://youtu.be/PMMc4VsIacU?si=lVbv70Wa70XI7dcC> (Best, Best, Best for learning what DFS is, simulation and pseudo code)
      1. Simulation (Time Related)
         1. <https://youtu.be/5Z7tnNZj7FY?si=riSYG9-v5IQJ0Uwp> (Good, taught in the class)
         2. <https://youtu.be/9w34hoqPvmY?si=jFcHmyFvg7nksTZW> (Good, taught in the class)
      2. **DFS; Kind of Edges** 
         1. Tree Edges
         2. Back Edges
         3. Cross Edges
         4. Forward Edges
   3. **Detect Cycle in Undirected graph Using BFS, DFS** 
      1. Undirected: <https://youtu.be/58Yv1zc1L1Q?si=dCYBqaaUn4jn4xHw>, and Directed: <https://youtu.be/WkTm8qu1lqU?si=0BL7c0a6zMxjci56> (Good, Taught in the class)
      2. <https://youtu.be/9twcmtQj4DU?si=6xXhHS4SjIl2tlMY> (Good)
   4. **Find the smallest/shortest path using BFS** 
      1. Simulation
         1. <https://youtu.be/XF6glBnDxd0?si=C1WKJs6OiCKdErp1> (Talk in very simple word, good)
         2. <https://youtu.be/UvxV6y0k6Vk?si=xYKTMoikj6j_MZAJ> (Chole)
         3. <https://youtu.be/hwCWi7-bRfI?si=0c4HCD4EDqpJud3X> (Oshadharon, Super, Super, taught what discussed in the class) and <https://youtu.be/CrxG4WJotgg?si=9AEwBdEsBnHLiood>
         4. <https://youtu.be/abIEXKFpLNE?si=ZX5grzOv3NFt2aLO> (Oshadharon, Super, Super, taught what discussed in the class)
         5. <https://youtu.be/WOV3LYhPews?si=oy-Ecp-ywGNdSq6a> (CP, good)
         6. <https://youtu.be/C4gxoTaI71U?si=Zs6vKG_BgNrrbhs6> (SUPER, super)
4. **Detecting Bipartite Graph**
5. **Topological Sort** (Need DFS) / **Directed Acyclic Graph - DAG**
   1. USE: DFS, VISITED array, STACK
   2. <https://youtu.be/3tkcfvCNtM8?si=R86IIgosvR8TpvRN> (Good, Good, 10/10)
   3. <https://youtu.be/5lZ0iJMrUMk?si=LAgwy0rXGQO-uVnn> and, Implement Topological Sort using Kahn’s Algorithm: <https://youtu.be/73sneFXuTEg?si=xwM2hMc1gjpebaIk> (Good, Good, Good, Oshadharon)
   4. Shortest Path in Directed Acyclic Graph - Topological Sort: <https://youtu.be/ZUFQfFaU-8U?si=ayTC6SjjD4GGAm1A> (Oshadharon, Super, Super, taught what discussed in the class) 🡪 Little Bit Advance.
   5. <https://youtu.be/T_boOrr0rvk?si=g6JwlcFErytvj9eg> (Good, 10/10) and, Implement Topological Sort using Kahn’s Algorithm: <https://youtu.be/6XmzL04mlgQ?si=7Ilb6fCkNl8QpLh3> (Good)
   6. Shortest Path in Directed Acyclic Graph: <https://youtu.be/P_bfy0LOU5g?si=YdBb9XC8qvYYEzN8> (Oshadharon, Super, Super, taught what discussed in the class) 🡪 Little Bit Advance.
   7. <https://youtu.be/mbE-LdQgWzY?si=s9enRy7j30nY3OY2> (taught MTD slide, SUPER)
   8. <https://youtu.be/iB0pYfUUctg?si=8W16kpDkTIWqoFc_> (Motamoti)
   9. <https://youtu.be/KqgPn8nshgU?si=IctNn7_AaifqIood> (Chole)
   10. <https://youtu.be/7J3GadLzydI?si=0zhFrkwsEgePbtNQ> (Good, 6/10)
   11. Code: <https://youtu.be/GYmq98CVm2c?si=-dU2eXtgS1ymoX0L> (Chole)
   12. <https://youtu.be/yYW6lQ1ajx4?si=a7EJYSpxlb0Bqrlb> (Motamoti, for knowing in short)
   13. <https://youtu.be/i9Uo7B1WiEE?si=M6qyazou31K9Q4nD> (5/10)
   14. <https://youtu.be/ddTC4Zovtbc?si=27FjPXVtJO967hcK> (Good, Good, 9/10)
6. **Find Strongly Connected Components from the graph** (Need DFS) -> **Kosaraju's Algorithm**
   1. Definition: Jei 2 jon er moddhe amra eta ber korbo, dekhbo je oi 2 jon ki ekjon arekjon er kache ashte pare naki, jodi 2 jon e 2 jon er kache jete pare, then we can say that they are strongly connected components.
   2. USE: 🡪 DFS, STACK, VISITED array 🡪 REVERSE THE GRAPH 🡪 POP FROM THE STACK, AND WRITE IT AS AN ANSWER.
   3. <https://youtu.be/LvM8Qi-IvqE?si=HVDtZP_O6MZiQRqY> (Good, 7/10)
   4. <https://youtu.be/R6uoSjZ2imo?si=5XBlB7KOy_rT3dpz> (Super, Super, Super, 1000/10)
   5. <https://youtu.be/RpgcYiky7uw?si=s2_D3yM2CGUrYqa1> (Good, 10/10)
   6. <https://youtu.be/ju9Yk7OOEb8?si=8KJJSvzS6IDj8yeU> (Good, 10/10)
   7. <https://youtu.be/Rs6DXyWpWrI?si=Hu-_xMsM-LRI1Yop> (Super, Super, Super, Excellent, 10000000000000/10, Covered Everything)
7. Weighted Graph and Shortest Path
8. Single-Source Shortest Path Problem
   * 1. Negative weighted cycle
     2. Relaxation
   1. Used to find the shortest path for weighted graph
      1. **Dijkstra’s Algorithm** (positive edge weights only) 🡪 O( E\*log(V) )
         1. Undirected: <https://youtu.be/bHolUy5sqq8?si=Av6BKB33D0a6j2SV> (Good, Good, Oshadharon, Super, Super) and Directed: <https://youtu.be/p0MUtpaw1ks?si=F8CAeFka1Up662GH> (Good, Good, Oshadharon, Super, Super)
         2. <https://youtu.be/rumoDMIU-U0?si=KU71nChJrAnajKeD> (Joss)
         3. For practice: <https://youtu.be/0nVYi3o161A?si=HDybXZNNJhDmKbAf> (Good, Good, similar to the 1&2 videos)
         4. For practice: <https://youtu.be/4UMlxaAFGSA?si=AMFhYyzOk-CRAe96> (Good, Good, similar to the 1&2 videos)
         5. <https://youtu.be/EFg3u_E6eHU?si=wXyuy5dwurOIzcP3> (Good, for knowing how it works)
         6. For Practice: <https://youtu.be/Gd92jSu_cZk?si=Uv0MjJT-fy1z4uGQ> and For Code: <https://youtu.be/_0s2e5SqhSk?si=YrKkgtWU8U-iSQT-> and For knowing Where it FAILS: <https://youtu.be/R3g2SSlyY_0?si=-tOebuCA5818y5jw> (Good, Good, Oshadharon, Super, Super)
         7. For Practice: <https://youtu.be/V6H1qAeB-l4?si=qeV6c1YxKlJFSrpx> (Good)
         8. For Practice: <https://youtu.be/Sj5Z-jaE2x0?si=R0h_WrW_aq9oVlsV> and Code: <https://youtu.be/t2d-XYuPfg0?si=52jVXVsOB-XkGOox> (Good)
         9. For Practice: <https://youtu.be/F3PNsWE6_hM?si=CEqpN2TIhEbXSZsI> (Good)
         10. For Practice: <https://youtu.be/dVUR3Rm6biE?si=WkDdUT2YmoLVLTE6> (Good)
         11. <https://youtu.be/XB4MIexjvY0?si=jwkFOhVGvmhes5Rr> (Good)
         12. <https://youtu.be/lAXZGERcDf4?si=ODnj1pMjOrEROp_D>
         13. <https://youtu.be/smHnz2RHJBY?si=Xar1Y6Ql8_mVpB65> (Good 9/10)
         14. For practice: <https://youtu.be/U5W8-gGblXs?si=6ZG8AigU4KUgU8Qe> and <https://youtu.be/Rv2RzVu9S9k?si=fy9NanCNz9PS8MwB> (Good)
         15. For practice: <https://youtu.be/r4U342MdMj0?si=IS9v8WAZlOXWffSM> and <https://youtu.be/ky_8BWwv_pY?si=u4STW_hLuOH-WYNi>
         16. For practice: <https://youtu.be/vpmY9_hV0lQ?si=0Rv5hn0t93CgnmdG>
         17. For practice: <https://youtu.be/bZkzH5x0SKU?si=XnuWIc4QXC18n3SN> (Good)
         18. <https://youtu.be/pVfj6mxhdMw?si=ZORCVppuFpcRiMWj> (Chole)
      2. **Bellman-Ford algorithm** (positive and negative edge weights)
         1. <https://youtu.be/2fZw1BtkoTk?si=bGbMpD0Aq-F3Jis8> (Good, Good, Oshadharon, Super, Super)
         2. <https://youtu.be/SiI03wnREt4?si=-iUSR-04bnE8Mv6J> (Awesome)
         3. For practice: <https://youtu.be/obWXjtg0L64?si=TzYLwTFDNceGeDgD> (Good)
         4. <https://youtu.be/ijpVpsmpJtQ?si=MObIjkssTplCVTD8>
         5. <https://youtu.be/KudAWAMiQog?si=TmTMtBYyrEVpPJYq>
         6. <https://youtu.be/FtN3BYH2Zes?si=UztckObwaNOPlq9p>
         7. <https://youtu.be/lyw4FaxrwHg?si=Hc5o2iFkNZbJysYB>
         8. <https://youtu.be/7bSbkIZ_wfc?si=By1YAJLI4xiF5GIE>
         9. <https://youtu.be/iTW2yFYd1Nc?si=xh4vimvo29mhQ-qd>
         10. <https://youtu.be/GkQAxgEm5qw?si=sDvg0EV0Q0DpO2lp>
         11. <https://youtu.be/FrLWd1tJ_Wc?si=KleQZlGcf4qQynEu>
         12. <https://youtu.be/LKfIjVZ6kg4?si=7Own_4JJDQpUmsk3>
         13. <https://youtu.be/T_jIM6_JrQM?si=CbXLC3ESkZthVBoP>
         14. <https://youtu.be/Z62R0cZ--tY?si=-sZmV0WRc83rH7U_>
9. Minimum Spanning Tree
   * 1. <https://youtu.be/CuB3-dmiSL0?si=x2AOUuXmsQV1Js2C> (Good, Good, Oshadharon, Super, Super)
     2. <https://youtu.be/4ZlRH0eK-qQ?si=lsSo2mGEA4pUbefQ> (Good)
     3. <https://youtu.be/ZSPjZuZWCME?si=m_sL8y0O-Mn6Mi0J> (Good, Good)
     4. <https://youtu.be/oh3ZOtVix8Y?si=cGSayGdpMZ9jSW7j> (Chole)
   1. **Prim** 
      1. <https://youtu.be/KwYMYX0a73k?si=-suEdms_1ua6HhLY> (Good, Good, Oshadharon, Super, Super)
      2. <https://youtu.be/mJcZjjKzeqk?si=qJlsp7e0m0DLm6rX> (Chole)
      3. <https://youtu.be/_KX8GDvRzBc?si=UYe9JKH8h4zNRwc0> (\*\*\*\*\*)
   2. **Kruskal Algorithm** 
      1. <https://youtu.be/Qt0tUIzoj8k?si=p4M7WvsMbklHSSDq> (Good, Good, Oshadharon, Super, Super)
      2. <https://youtu.be/DMnDM_sxVig?si=BXvQHVW4gm4-eLxv> (Good, Good)
10. Greedy Algorithms
    1. Interval scheduling problem
    2. Scheduling all intervals problem
    3. Fractional knapsack problem
       1. Extension part: 0/1 knapsack problem
    4. Coin changing problem is
    5. What problems can be solved by greedy approach?
    6. Huffman Coding (Example: Zip)
11. Dynamic Programming
    1. 0/1 knapsack problem
    2. Longest Common Subsequence - LCS
       1. Recursive Relation
       2. Table fill-up; \*for finding the length, and \*printing the longest common subsequence
       3. Pseudo Code
12. P – NP

Hello everyone,

Here is the full syllabus for your final. The first 4 videos were from online class. The 5th one is recorded by me and the Final video (topic -P vs NP) is shared with all students of 221. So, these 6 recordings are enough for your final exam. Start studying properly. So, no official class from tomorrow, I will arrange some online discussion sessions

where you will be able to ask questions after studying all the topics.

Recording-1 (Topological Sort + SCC): <https://drive.google.com/file/d/13IH3p0ZyQ1YRmDfjwFCmE-IfB1jr0V1n/view?usp=drive_link>

Recording-2 (MST) : <https://drive.google.com/file/d/1ZgE3uCuubJiieY1U2AOoeWqkEenGlyrd/view>

Recording-3 (Dijkstra + Bellman Ford): <https://drive.google.com/file/d/1Fp36mlhbI-X7n9F3PVZdFaJjph4jD_Od/view?usp=sharing>

Recording-4 (Greedy- Huffman coding + Fractional Knapsack): <https://drive.google.com/file/d/1Z5U9ChWdg0ZpJJwuEXeYd8t_3zcLtQTv/view?usp=sharing>

Recording-5 (DP - LCS + Binary Knapsack)

<https://drive.google.com/file/d/19qKbJHUNxDVCOku1uFaNiG-JTbeuLCCz/view?usp=drive_link>

(Part2) - <https://drive.google.com/file/d/1nDwYGbRy2F6HW3qfmPVDBoD5DqV5I2i8/view?usp=drive_link>

Recording-6 (P vs NP) -

<https://bracuacbd-my.sharepoint.com/personal/fakhruddin_gazzali_bracu_ac_bd/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Ffakhruddin%5Fgazzali%5Fbracu%5Fac%5Fbd%2FDocuments%2FRecordings%2FP%2DNP%20Discussions%20%28Lectures%20%2B%20Video%29&amp;ga=1>