**MY CLASS RECORDING PLAYLIST:** [**https://youtube.com/playlist?list=PLgU-6-TtNPAzpauVSPWpKFbmu5Rbc2dEQ&si=PAzhNHH1uGKTAaTf**](https://youtube.com/playlist?list=PLgU-6-TtNPAzpauVSPWpKFbmu5Rbc2dEQ&si=PAzhNHH1uGKTAaTf)

**[dijkstra class is missing, giving some supplementary links for that]**

**For dijkstra algo:** [**Dijkstra's Shortest Path Algorithm | Graph Theory**](https://www.youtube.com/watch?v=pSqmAO-m7Lk&t=199s&ab_channel=WilliamFiset)

**[you can ignore the optimization parts and just focus on the simulation for now]**

**\*\*\*\*\*GRAPH TRAVERSAL\*\*\*\*\*\***

**Topic:GRAPH:** [**6. Graph [HFN] Supplementary videos**](https://docs.google.com/document/d/1sxSvO5w_Au8mpINOR6Y90dFo6x2Uze6gCqOTV9BX1os/edit?usp=sharing)

**Basic**

1. **Graph basic: you need to know that in order to do the tasks**

**Traversal algos [you can refer to my recordings Lecture 1 2]**

1. **BFS: You can find in the class recordings + this slide** [**6.1 Graph[BUX slide 8].pdf**](https://drive.google.com/file/d/14CuWxAgcVLdMpRNnTuJyPkmVSyReScN4/view?usp=drive_link)
2. **DFS: Recordings of my class +** [**6.1 Graph[BUX slide 8].pdf**](https://drive.google.com/file/d/14CuWxAgcVLdMpRNnTuJyPkmVSyReScN4/view?usp=drive_link) **this slide**

**Traversal comparison:** [**A Comparison of Pathfinding Algorithms**](https://www.youtube.com/watch?v=GC-nBgi9r0U&t=213s) **[SKIP A\* Algorithm for now]**

**Implementational algos: [refer to my recordings Lecture 2 3]**

**[Can refer to the slide : 6.1, 6.2, 6.3, 6.4 from our shared link:** [**https://drive.google.com/drive/folders/1EuJTeoAiPXXtdCI5O3k48XDK5\_z4O7xU?usp=drive\_link**](https://drive.google.com/drive/folders/1EuJTeoAiPXXtdCI5O3k48XDK5_z4O7xU?usp=drive_link)**]**

1. **BFS: shortest path in an unweighted graph (refer to the slide above), level order demonstration since it explores level by level so any problem regarding to those, also same slide. For identifying shortest path sequence maintain a parent pointer.**
2. **DFS: edge classification:** [**DFS - Types of Edges | Edge Classification | Tree Edge, Back Edge, Forward Edge, Cross Edge**](https://www.youtube.com/watch?v=Y78KivF-hm0)

**How to find Cycle in graph: (DFS method)** [**Detect cycle in a directed graph**](https://www.youtube.com/watch?v=0dJmTuMrUZM&t=3s)

1. **Topological sort: (DFS)** [**Topological Sort Algorithm | Graph Theory**](https://www.youtube.com/watch?v=eL-KzMXSXXI&t=666s)**(BFS/ Kahn’s ALgo)**

**Implementation of top sort:** [**Shortest/Longest path on a Directed Acyclic Graph (DAG) | Graph Theory**](https://www.youtube.com/watch?v=TXkDpqjDMHA)

1. **Strongly Connected Components:**

**1**[**Kosaraju Algorithm | Strongly connected components in a graph**](https://www.youtube.com/watch?v=Rs6DXyWpWrI&t=220s)

**2**[**Tarjans strongly connected components algorithm**](https://www.youtube.com/watch?v=ZeDNSeilf-Y&t=1s) **[OPTIONAL]**

**\*\*\*\*\*PATH FINDING/ SHORTEEST PATH ALGO\*\*\*\*\***

**FOR DIJKSTRA SIMULATION FOLLOW THIS VIDEO:**

[**Graph Data Structure 4. Dijkstra’s Shortest Path Algorithm**](https://www.youtube.com/watch?v=pVfj6mxhdMw&t=479s&ab_channel=ComputerScience)

**Single source shortest path algorithms: [USE THESE VIDEOs]**

[**Dijkstra's Shortest Path Algorithm | Graph Theory**](https://www.youtube.com/watch?v=pSqmAO-m7Lk)

**[Ignore the optimization parts]**

[**Graph Data Structure 4. Dijkstra’s Shortest Path Algorithm**](https://www.youtube.com/watch?v=pVfj6mxhdMw&t=573s&ab_channel=ComputerScience)

**Negative weight cycle issue:** [**Why Dijkstra's Algorithm Doesn't Work with Negative Weights**](https://www.youtube.com/watch?v=eXPw7BBMFNk&ab_channel=CarlthePerson)

**To handle that, we use: [just know the general idea of bellman]**

[**Bellman Ford Algorithm | Shortest path & Negative cycles | Graph Theory**](https://www.youtube.com/watch?v=lyw4FaxrwHg&t=224s)

**\*\*\*\*\*GREEDY ALGO\*\*\*\*\*\***

1. **Huffman coding[Greedy Algo]: [refer to my recordings lecture 6]**

[**7. Huffman Coding**](https://docs.google.com/document/d/1MfVQeWIdhVrcgxKPhTzlPYrMYRue-6WxupWc00u-S8w/edit?usp=drive_link) **[all the links are given in this doc]**

1. **Fractional Knapsack [not dynamic]**[**Knapsack Problem | Greedy Method | Data structure & Algorithm | Bangla Tutorial**](https://www.youtube.com/watch?v=D8x5yIfjr5Q&ab_channel=FarhanHossan)

**\*\*\*\*\*\*MINIMUM SPANNING TREE\*\*\*\*\*\***

**Minimum Spanning Tree:** [**6.4 Dijkstra+ MST.pdf**](https://drive.google.com/file/d/1PDNW7cZaeBRBftnYA5RcY2qZoTdD9ueR/view?usp=drive_link) **[SLIDE]**

**[refer to my recordings Lecture 5 , 6]**

1. **Prims Algo:** [**Prim's Minimum Spanning Tree Algorithm | Graph Theory**](https://www.youtube.com/watch?v=jsmMtJpPnhU&ab_channel=WilliamFiset)
2. **Kruskal Algo:**[**Union Find Kruskal's Algorithm**](https://www.youtube.com/watch?v=JZBQLXgSGfs&ab_channel=WilliamFiset)
3. **Prims vs kruskal:** [**https://www.baeldung.com/cs/kruskals-vs-prims-algorithm**](https://www.baeldung.com/cs/kruskals-vs-prims-algorithm)

**\*\*\*\*\*\*DYNAMIC PROGRAMMING\*\*\*\*\*\***

**Dynamic Programming: [refer to my recordings Lecture 7,8]**

**01 knapsack:**

[**0/1 Knapsack Problem Dynamic Programming**](https://www.youtube.com/watch?v=8LusJS5-AGo&t=815s&ab_channel=TusharRoy-CodingMadeSimple)

**LCS:** [**4.9 Longest Common Subsequence (LCS) - Recursion and Dynamic Programming**](https://www.youtube.com/watch?v=sSno9rV8Rhg&t=1130s&ab_channel=AbdulBari)

**\*\*\*\*\*\*\*\*PvsNP\*\*\*\*\*\*\*\* [Just general ideas/intro is needed]**

**P vs NP problem:[USE THESE VIDEOs+slides]**

1. [**https://bracuacbd-my.sharepoint.com/:f:/g/personal/fakhruddin\_gazzali\_bracu\_ac\_bd/EkuceTu\_illFufQaAcqawGcBfgfdzpODwUFjgGUcBQxnZA?e=0wmLdo**](https://bracuacbd-my.sharepoint.com/:f:/g/personal/fakhruddin_gazzali_bracu_ac_bd/EkuceTu_illFufQaAcqawGcBfgfdzpODwUFjgGUcBQxnZA?e=0wmLdo)
2. [**NP.pdf**](https://drive.google.com/file/d/1H-hcL4cDZMKq8Ds1JM-ipJD7adjNcmd6/view?usp=share_link)

**ALL THE RELEVANT BUX SLIDES ARE GIVEN ALONG WITH THIS DOC IN THIS LINK:** [**https://drive.google.com/drive/folders/1EuJTeoAiPXXtdCI5O3k48XDK5\_z4O7xU?usp=drive\_link**](https://drive.google.com/drive/folders/1EuJTeoAiPXXtdCI5O3k48XDK5_z4O7xU?usp=drive_link)