**Class recording (before mid): covering Graph representation, BFS, DFS, Shortest path, Edge classification.** [**Class recording link**](https://youtu.be/wEthCHma_7w?feature=shared)

**\*\*ALL CLASS RECORDING PLAYLIST FOR ONLINE SESSIONS OF THIS SEMESTER:**

[**ONLINE CLASSES HFN**](https://www.youtube.com/playlist?list=PLgU-6-TtNPAzpauVSPWpKFbmu5Rbc2dEQ)**\*\***

**Edge classification supplementary:** [DFS - Types of Edges | Edge Classification | Tree Edge, Back Edge, Forward Edge, Cross Edge](https://www.youtube.com/watch?v=Y78KivF-hm0)

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

**Slide:** [Graph[BUX].pdf](https://drive.google.com/file/d/14CuWxAgcVLdMpRNnTuJyPkmVSyReScN4/view?usp=drive_link)

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

**Topological sort: (DFS)** [Topological Sort Algorithm | Graph Theory](https://www.youtube.com/watch?v=eL-KzMXSXXI&t=666s)

**(BFS/ Kahn’s ALgo)** [Topological Sort | Kahn's Algorithm | Graph Theory](https://www.youtube.com/watch?v=cIBFEhD77b4&t=521s)

**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)

**Single source shortest path algorithms:**

1. [Dijkstra's Shortest Path Algorithm | Graph Theory](https://www.youtube.com/watch?v=pSqmAO-m7Lk)
2. [Bellman Ford Algorithm | Shortest path & Negative cycles | Graph Theory](https://www.youtube.com/watch?v=lyw4FaxrwHg&t=224s)
3. [Shortest/Longest path on a Directed Acyclic Graph (DAG) | Graph Theory](https://www.youtube.com/watch?v=TXkDpqjDMHA)

**EXTRA CLASS: Tuesday 5th Dec [7.30 to 9.30]**

**Assignments: 3 [S1, S2, S3] s1: BFS,DFS,S1**

**Quiz: S1, S2/S3**

**Syllabus:**

1. **Single source shortest path Algo: Dijkstra, Bellman-Ford [1]**
2. **Greedy Algo: MST(Minimum Spanning Tree: Prims/Kruskal(DSU)) [2]**
3. **Huffman Encoding.**
4. **Dynamic Programming: Knapsack + LCS [2]**
5. **P vs NP [1]**