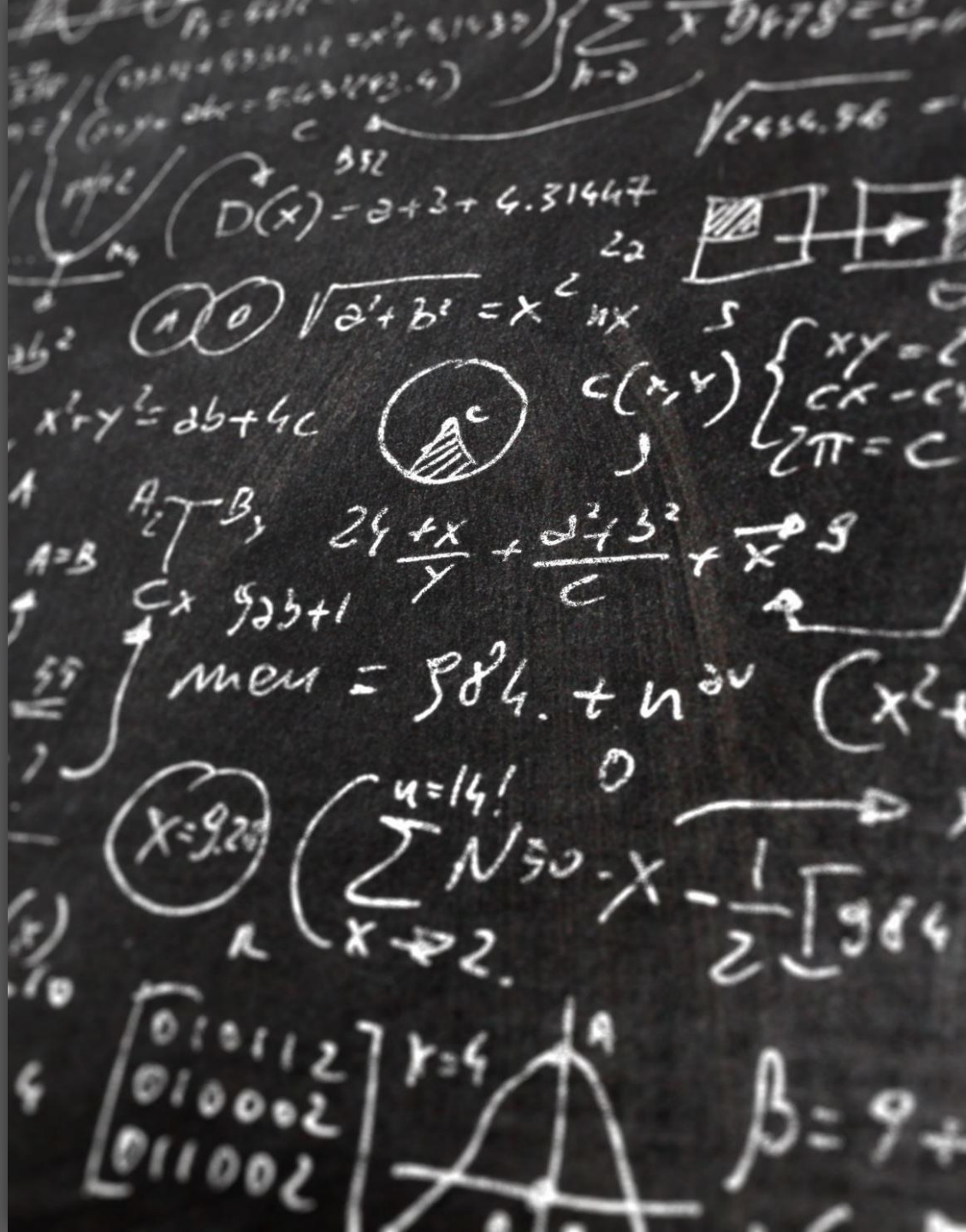
L1 MODULE 4

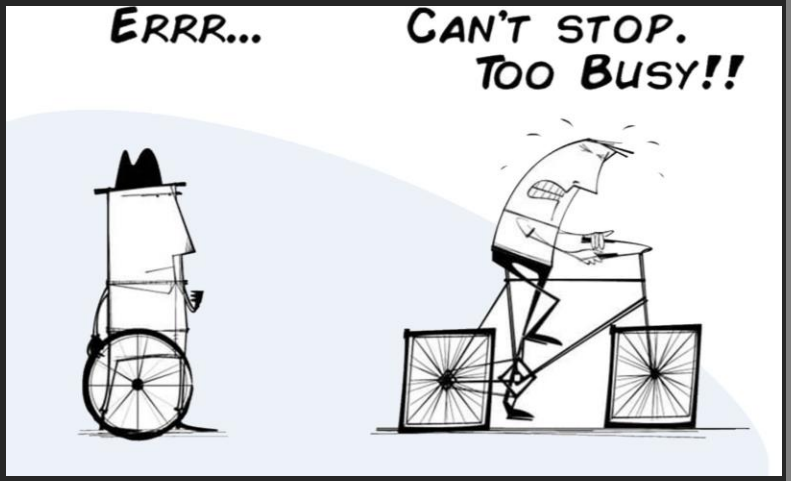
TABLE OF CONTENT

What is Technical Debt?

- Originally coined by Ward Cunningham(Inventor of Wiki).
- According to Trey Huffine, "Technical Debt is any code added now that will take more work to fix at a later time - typically with the purpose of achieving rapid gains."
- Shaun McCormick's definition of technical debt focuses more on the consequences in the long term, "I view technical debt as any code that decreases agility as the project matures.
- According to Gaminer, ""Technical debt happens when you take shortcuts in writing your code so that you achieve your goal faster, but at the cost of uglier, harder to maintain code".



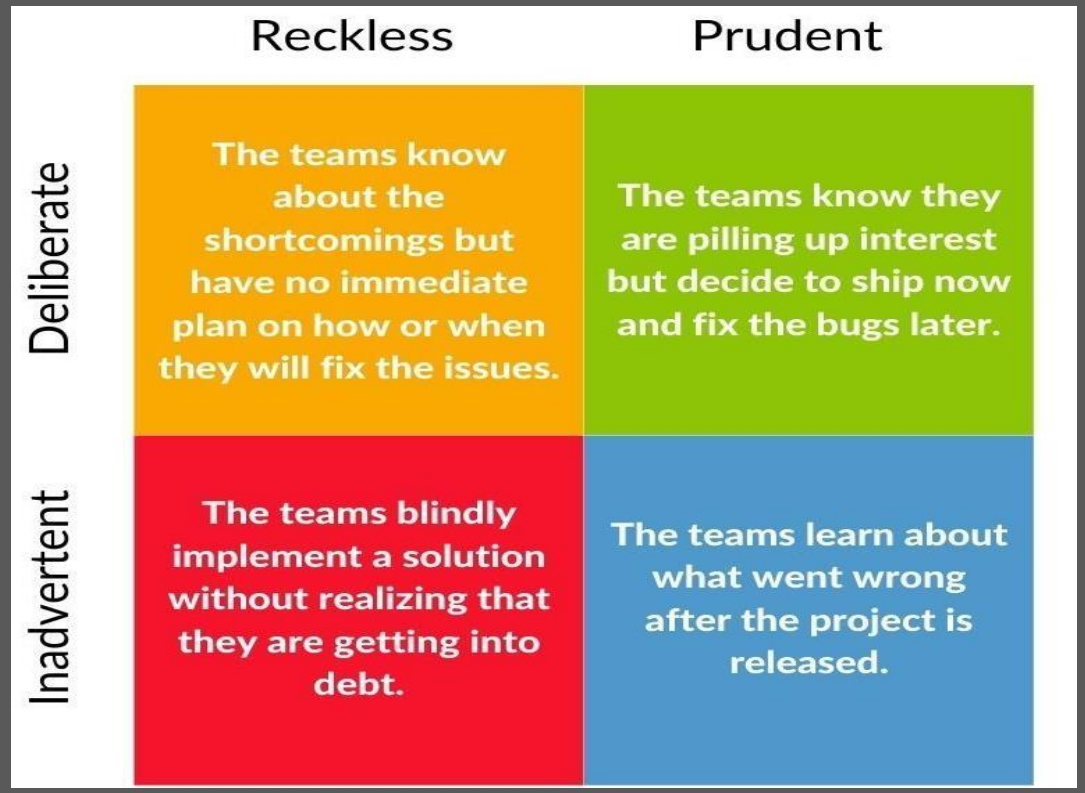
What it is not?



- A mess is not a technical debt. A mess is just a mess.
- A mess is always based on laziness and unprofessionalism and has no chance of paying off in the future. A mess is always a loss.
- But Technical debts are made based on real project constraints, may be a bit risky but beneficial.
- Technical Debt can be defined as " The consequences of software development actions that intentionally or unintentionally prioritize client value and/or project constraints such as delivery deadlines, over more technical implementation, and design considerations…”

Types of Technical Debt

According to Martin Fowler depending upon whether strategic or not



Depending upon nature of the debt

- Architecture Debt
- Build Debt
- Code Debt
- Defect Debt
- Design Debt
- Documentation Debt
- Infrastructure Debt
- People Debt
- Process Debt
- Requirement Debt
- Service Debt
- Test Automation Debt
- Test Debt

How To Measure?

Code Complexity Metrics (Cyclomatic Complexity, Maintainability Index)

Code Churn (Frequent changes in the same files)

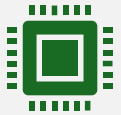
Defect Density (Bugs per 1,000 lines of code)

Test Coverage (Percentage of code covered by tests)

Static Code Analysis (Tools like SonarQube, CodeScene)

MANAGING STRATGIES

Refactoring & Code Reviews – Improve code quality iteratively.



Automated Testing & CI/CD – Reduce regression issues.

Documentation & Standards – Maintain best practices.

Prioritization & Debt Reduction Plan – Address high-impact areas first.

BUSINESS PROCESS AUTOMATION

What is BPA?

It leverages technology to automate operations and functions within a corporation.

Converts traditionally manual procedures into automated processes throughout the entire entity.

Enhance operational efficiency, minimize errors, standardize processes, and liberate employees to concentrate on more strategic, value-added activities.

Some Examples of Business process automation (BPA) include Employee Onboarding, Purchase orders, etc.

Why BPA?

Enhanced Efficiency

Increased Productivity

Substantial Cost Savings

Improved Accuracy and Consistency

Better Scalability

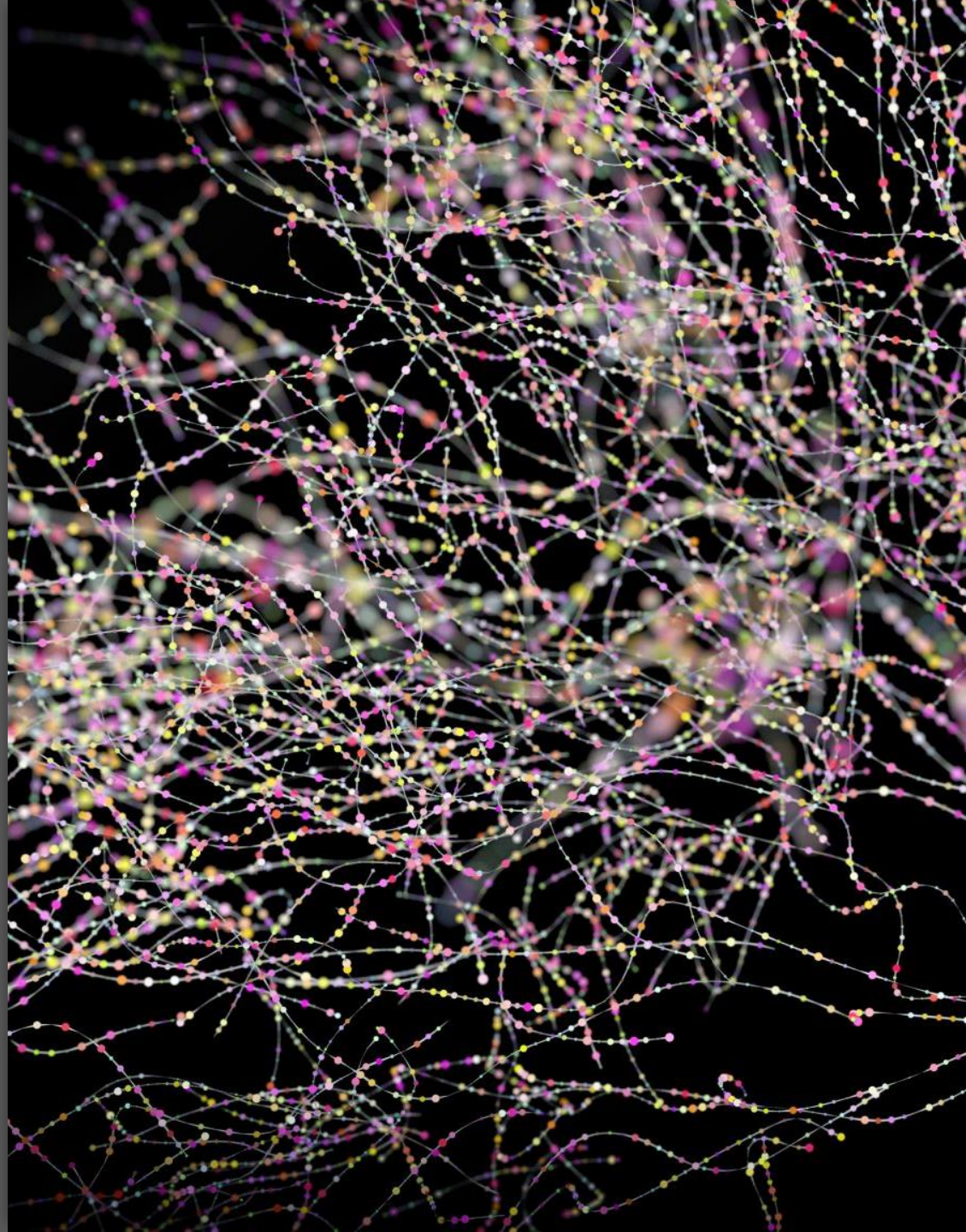
Increase customer satisfaction

How to automate Business Process?

- Identify Tasks to Be Automated
- Outline Business Goals
- Select the Right Tool
- Change Management
- Monitor and Measure

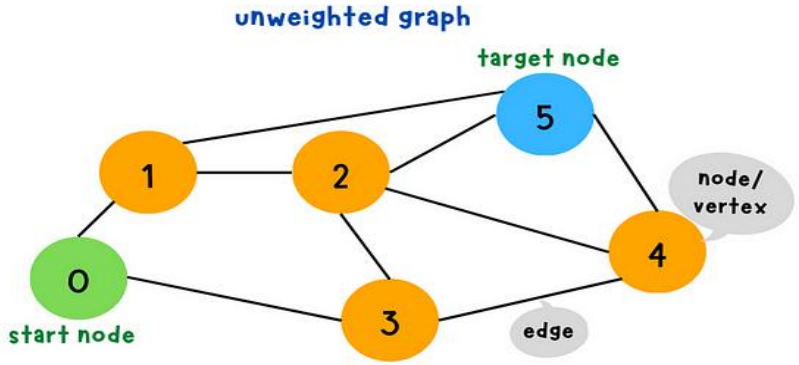
GRAPH ALGORITHMS

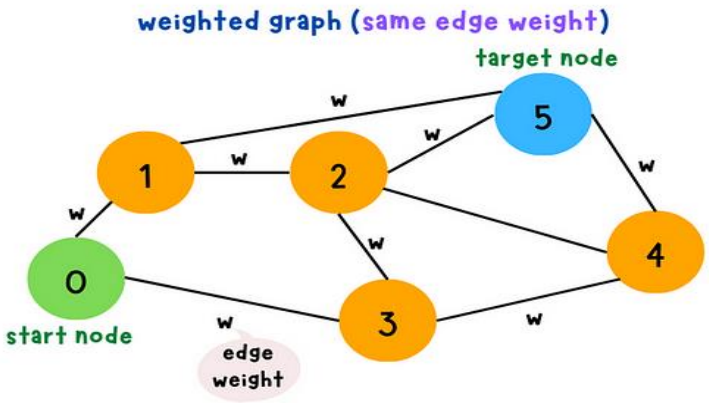
- BFS
- DFS
- TOPOLOGICAL SORT
- DIJKSTRA'S ALGORITHM



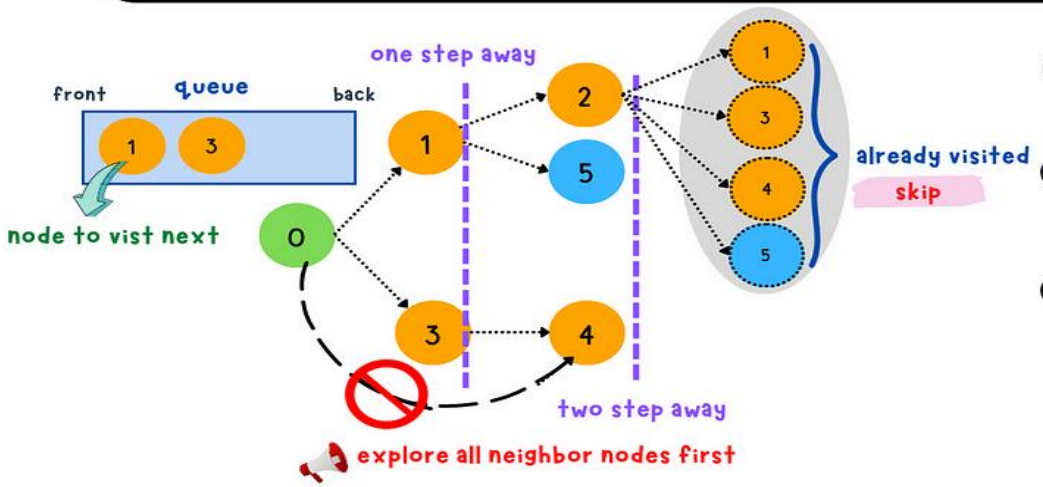
Breadth-First Search(BFS) Algorithm

unweighted graph or weighted graph having the same edge weight

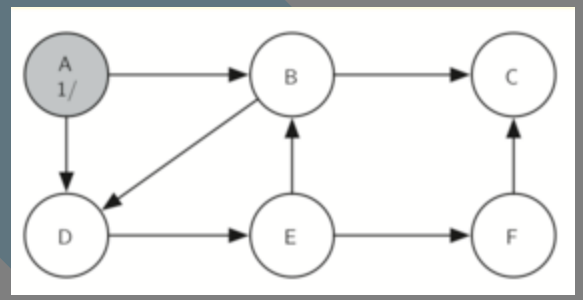


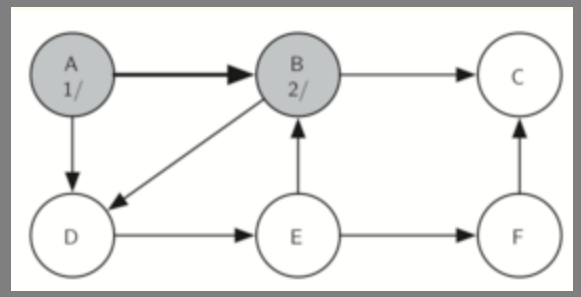


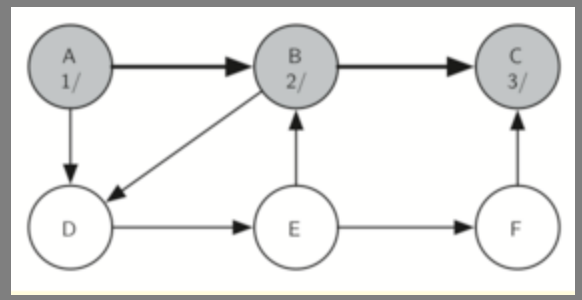
search the shortest path between the start node and the target node

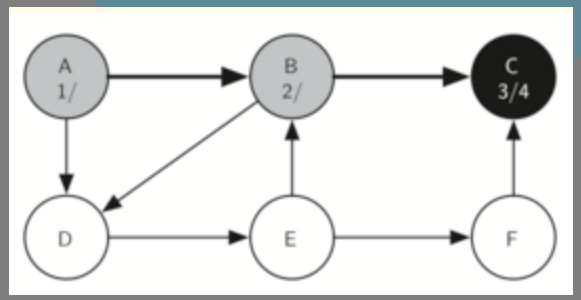


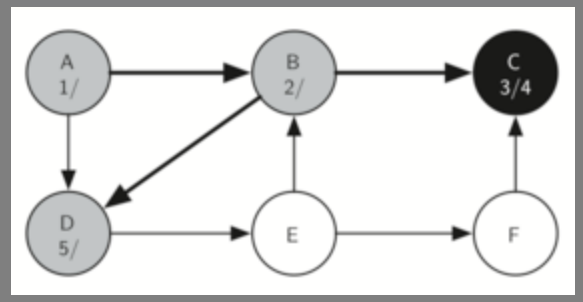
- search layer by layer
- use queue to store nodes to visit next
- every node and edge is visited once

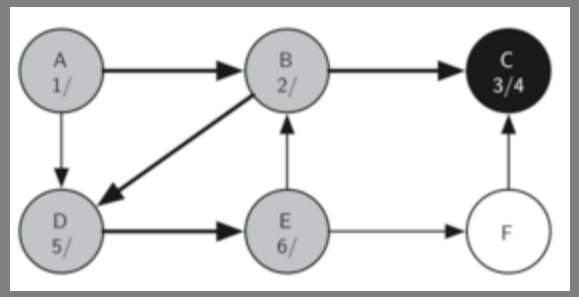


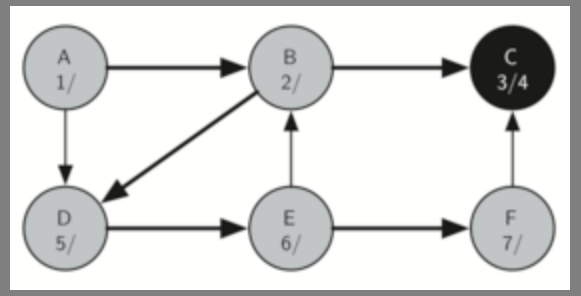


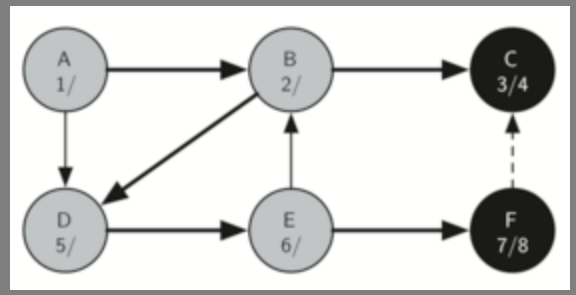


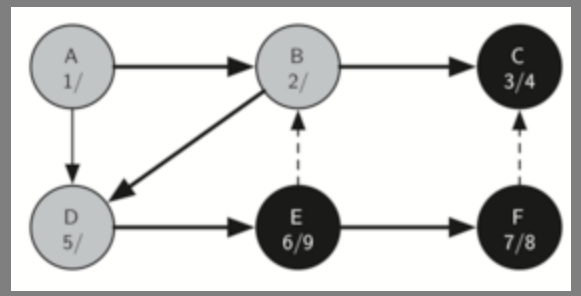


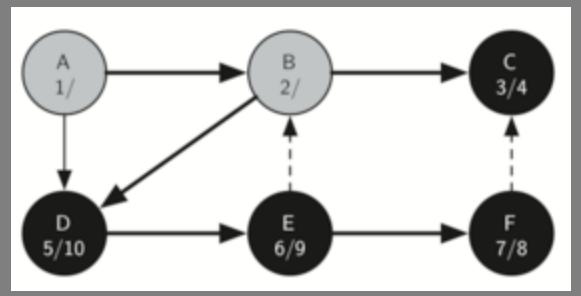


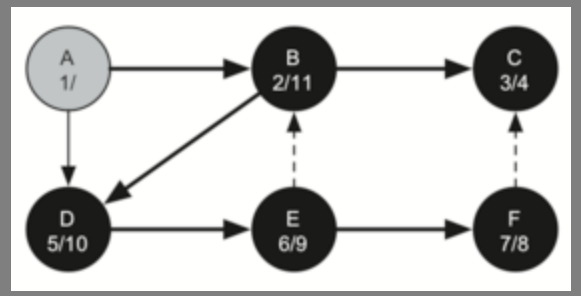


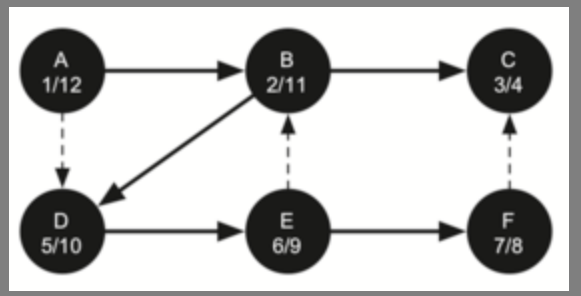




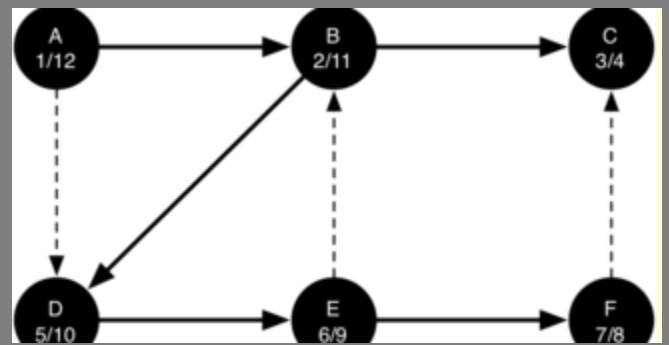


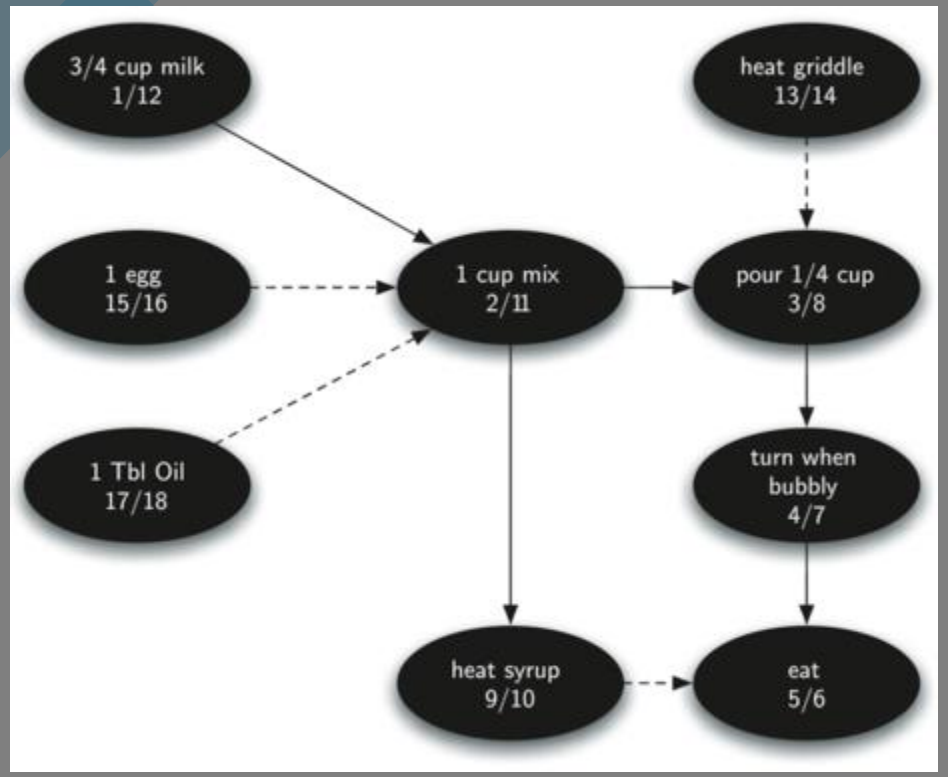






DFS ALGORITHM

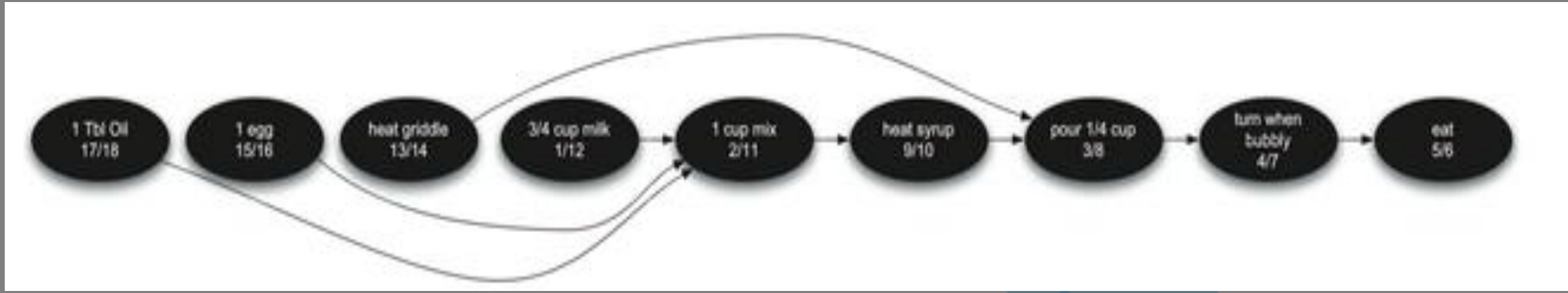




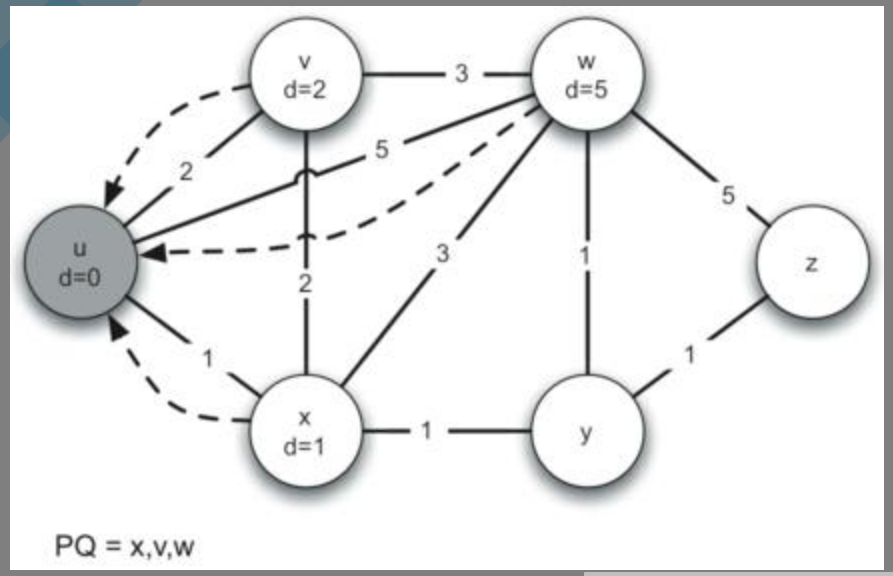
TOPOLOGICAL SORT

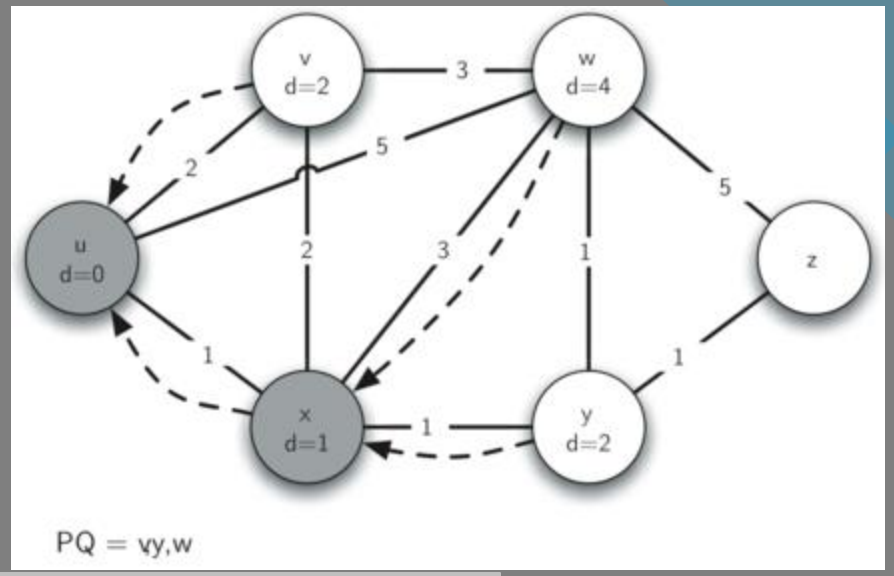
The topological sort is a simple but useful adaptation of a depth first search. The algorithm for the topological sort is as follows:

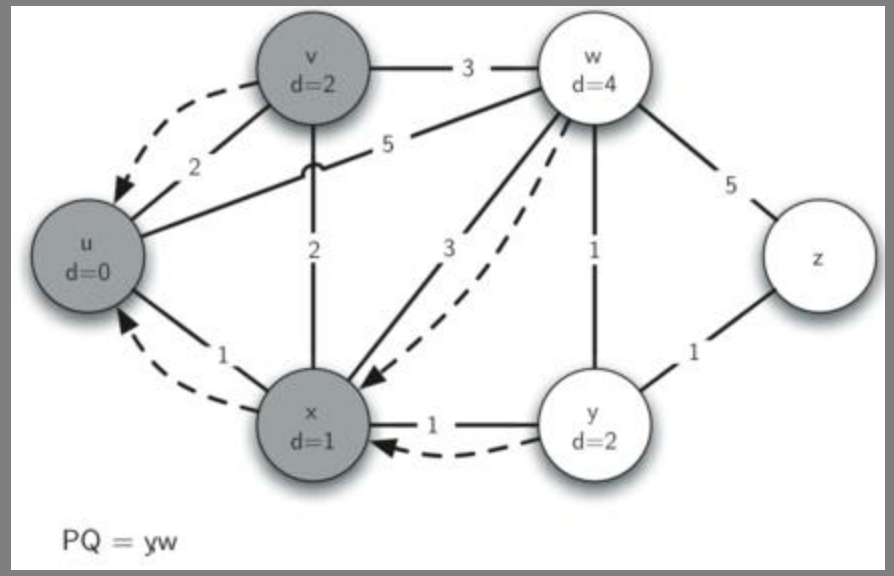
1. Call `dfs(g)` for some graph `g`. The main reason we want to call depth first search is to compute the finish times for each of the vertices.
2. Store the vertices in a list in decreasing order of finish time.
3. Return the ordered list as the result of the topological sort.

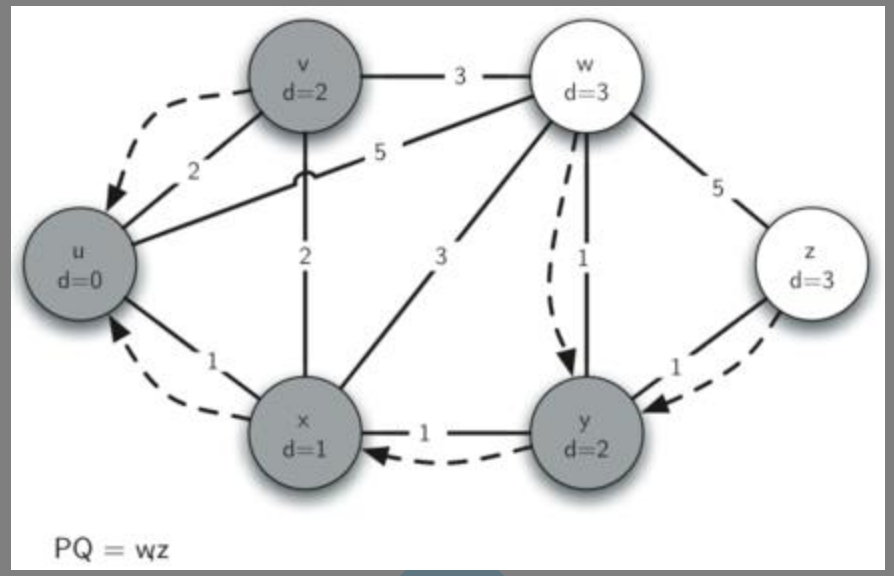


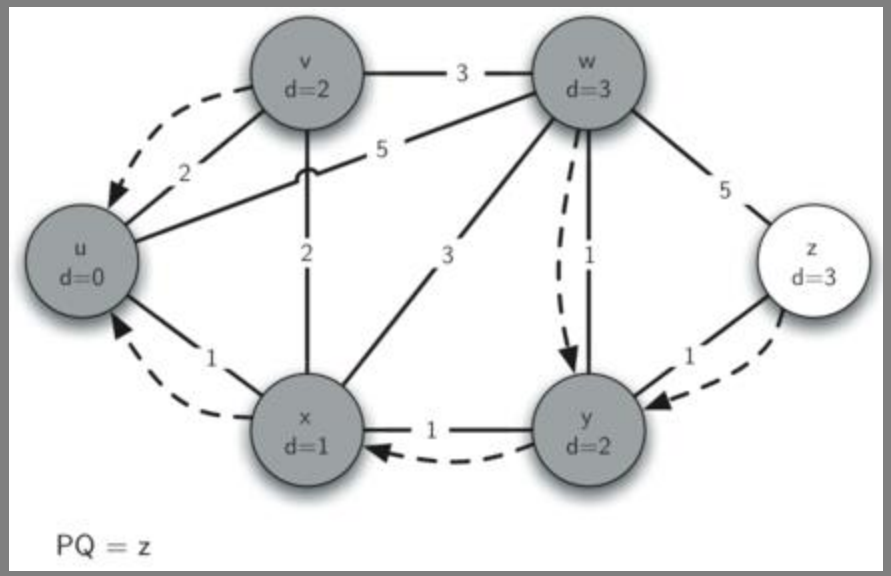
DIJKSTRA'S ALGORITHM

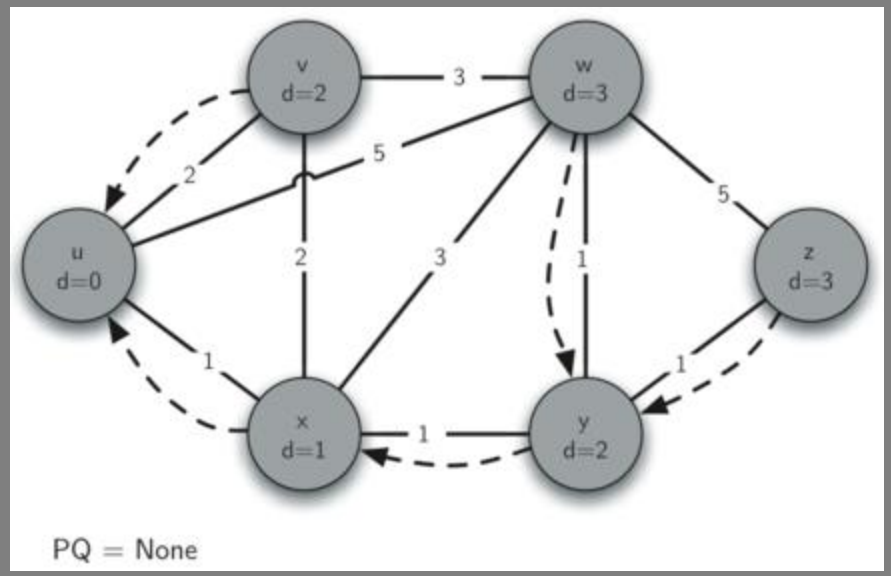












ASSIGNMENT LINK

Presented By:

Gopal Ranjan