

# Introduction to Fundamental Data Structures and Graph Algorithms

Presenter: Fandel Lin

12/4/2025

# Outline

- Queue
- Stack
- Breath-first algorithm
- Depth-first algorithm
- Topological sorting

# Queue

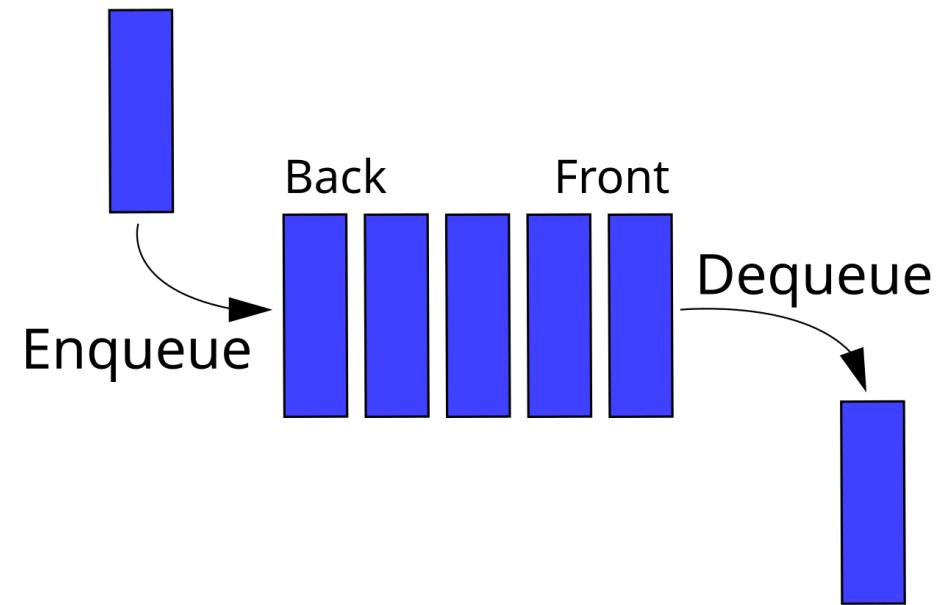
# Have you ever had the experience of

- Lining up to buy tickets at a movie theater



# Queue

- Data structure
  - Data storage format for efficient access
- FIFO (First in, first out)
  - Enqueue/ Dequeue
- Applications: **Preserve order**
  - e.g., printers



# Stack

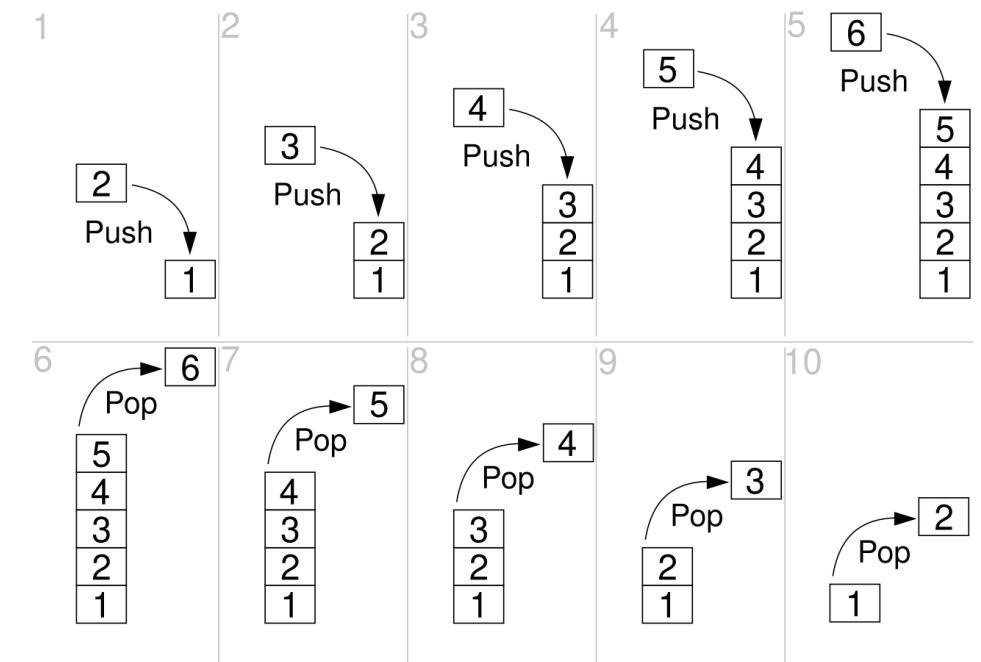
# Have you ever had the experience of

- Having a pile plates to take or wash



# Stack

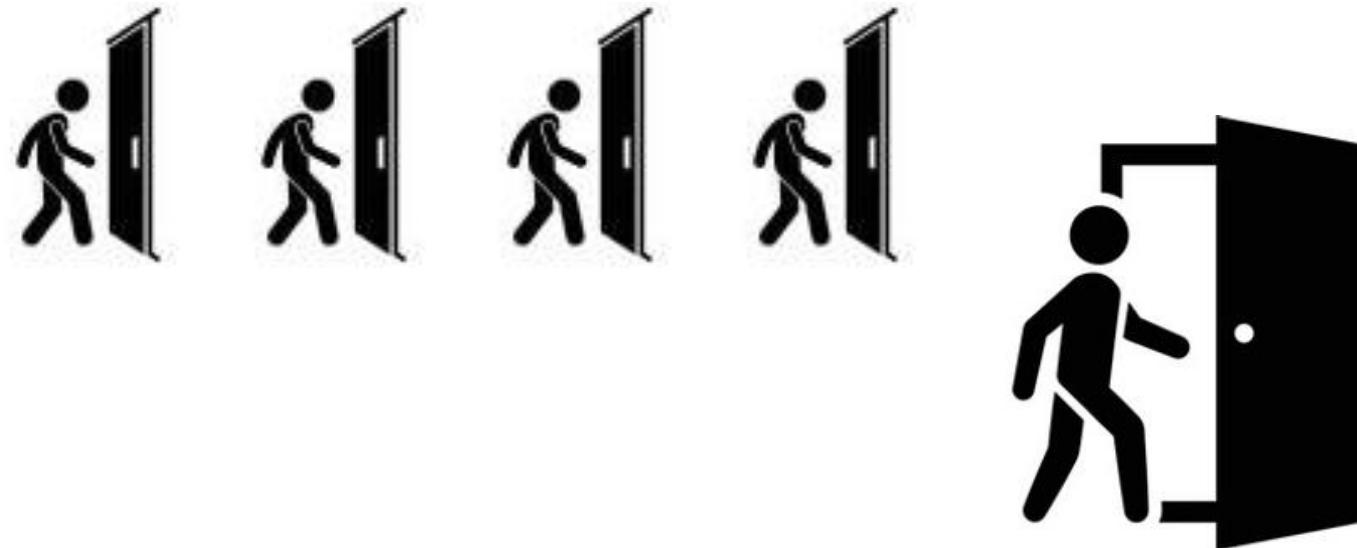
- Data structure
  - Data storage format for efficient access
- LIFO (Last in, first out)
  - Push/ Pop
- Applications: **Undo mechanisms**
  - E.g., browser history



# Breadth-First Algorithm

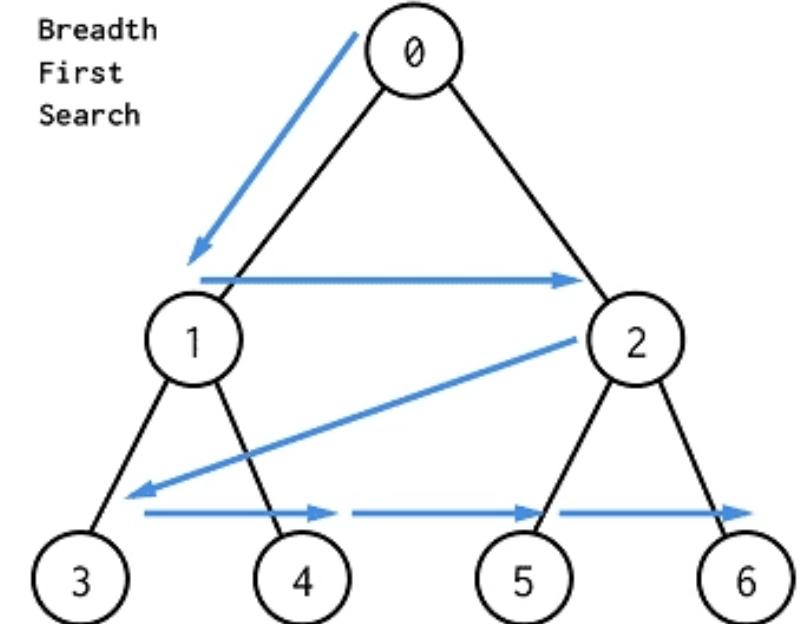
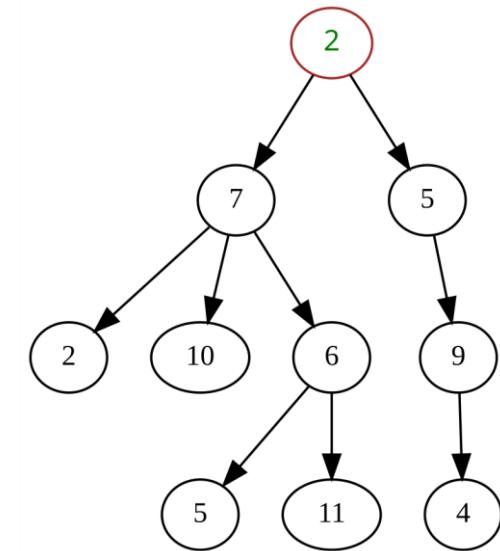
# Have you ever had the experience of

- Exploring doors to find the correct room in a hotel



# Breadth-First Algorithm

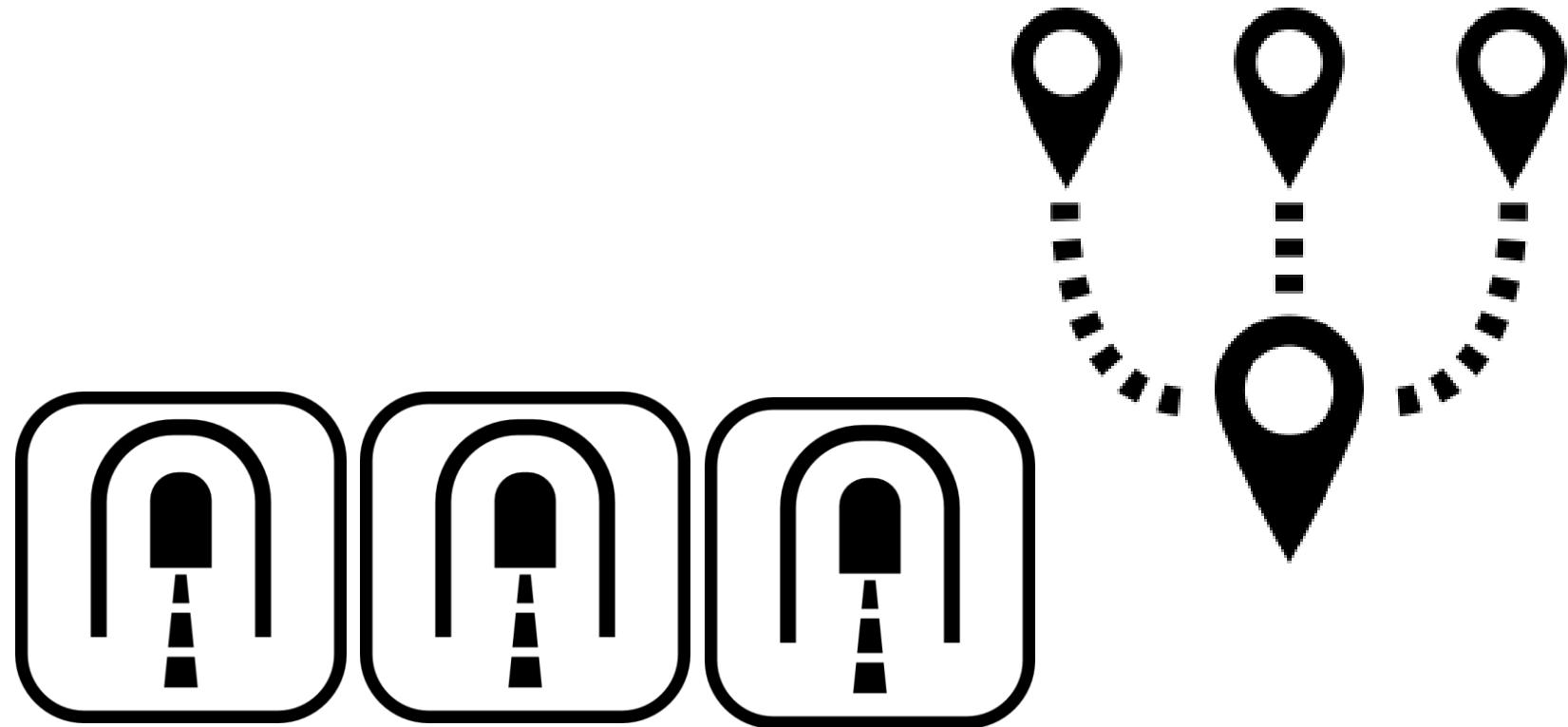
- Algorithm for searching a **tree** data structure
  - Tree: a set of connected nodes
- Finding the shortest path
  - Explore immediate neighbors before moving on
  - Use **queue** (FIFO) to keep track of the next step
- Applications: **Level-order traversal**



# Depth-First Algorithm

# Have you ever had the experience of

- Fully exploring one tunnels at a time when diving deep into a cave



# Depth-First Algorithm

- Algorithm for searching a **tree** data structure
  - Tree: a set of connected nodes
- Finding paths in deep structures
  - Explore as far as possible before backtracking
  - Use **stack** (LIFO) to keep track of the next step
- Applications: **Connection checking**

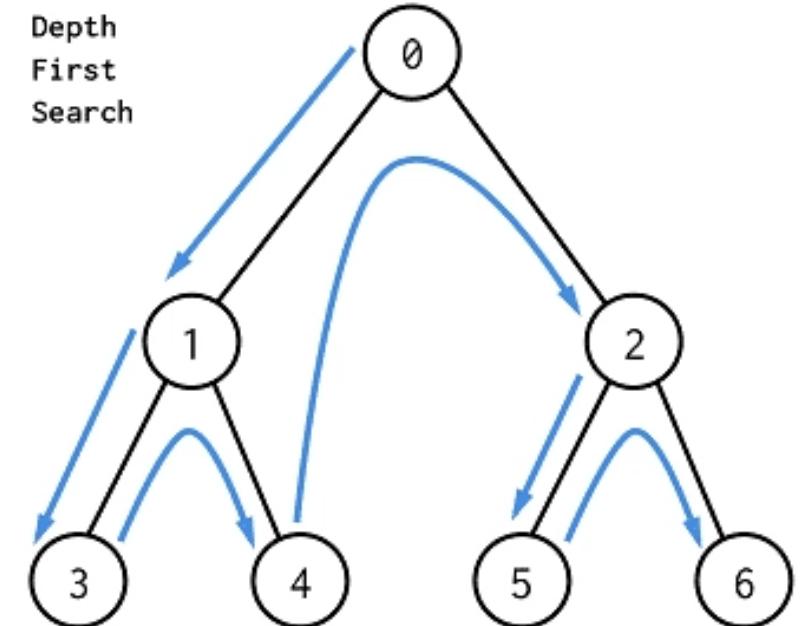
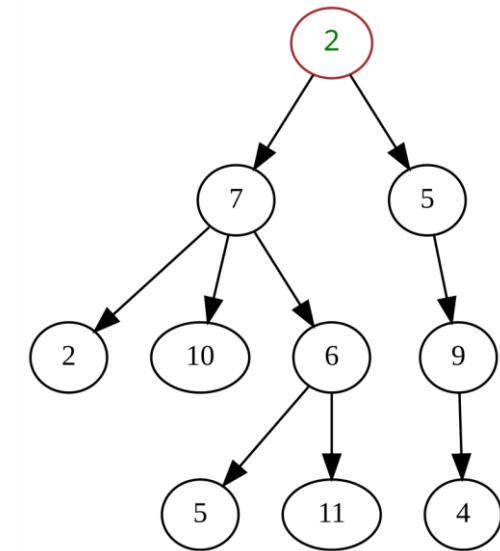


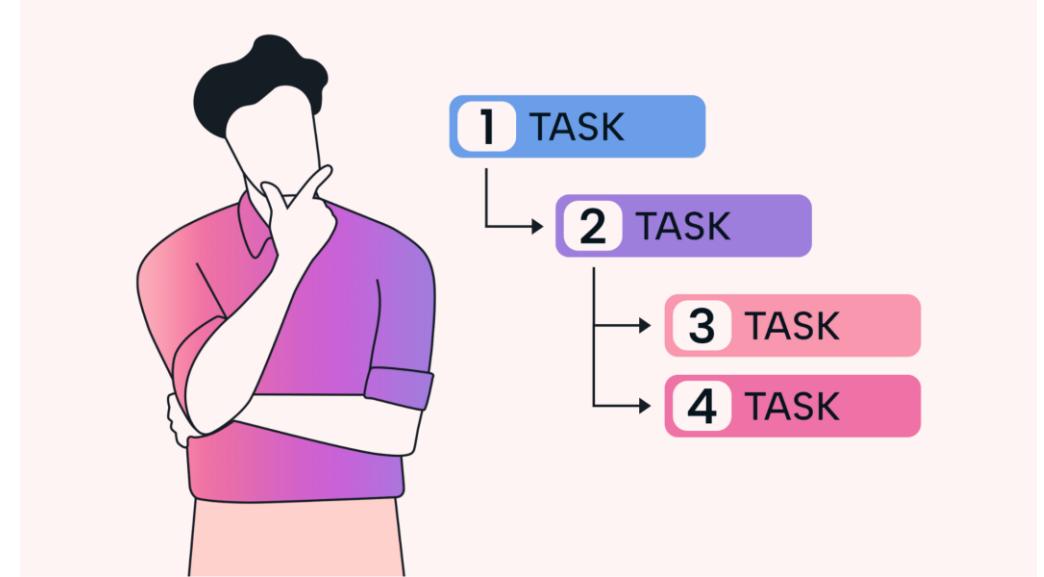
Image: [https://en.wikipedia.org/wiki/Tree\\_\(abstract\\_data\\_type\)#/media/File:Tree\\_\(computer\\_science\).svg](https://en.wikipedia.org/wiki/Tree_(abstract_data_type)#/media/File:Tree_(computer_science).svg)

Image: <https://dev.to/danimal92/difference-between-depth-first-search-and-breadth-first-search-6om>

# Topological Sorting

# Have you ever had the experience of

- Planning tasks in an order where one depends on another



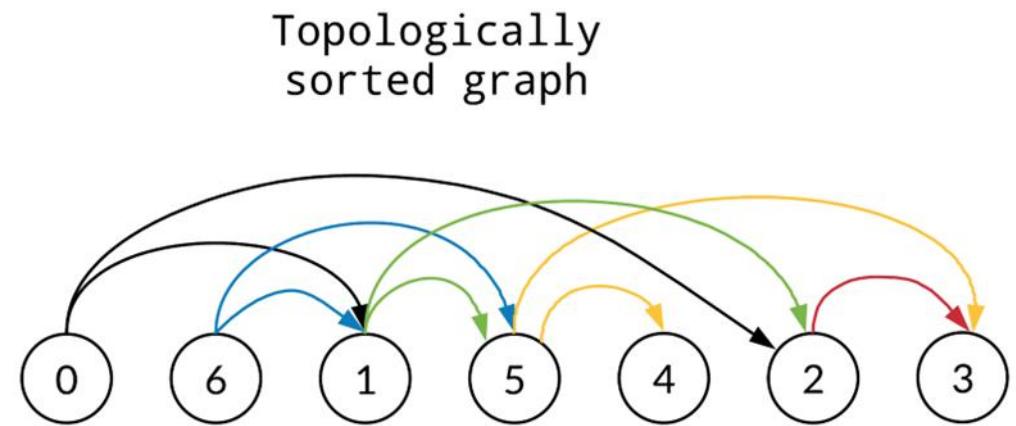
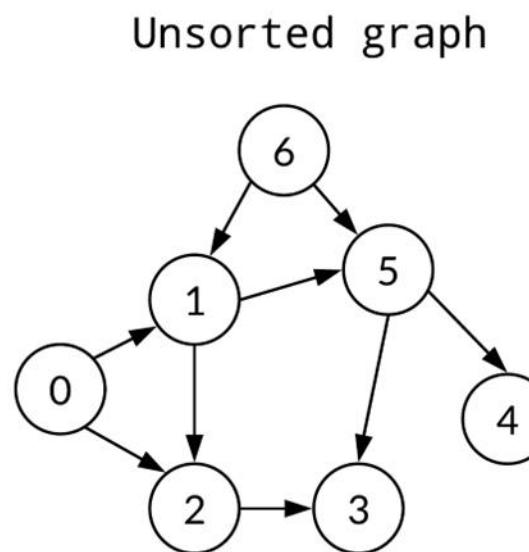
Topological ordering

# Topological Sorting

- Arranges the nodes of a **Directed acyclic graph (DAG)** in a linear order
  - DAG: a directed graph with no directed cycles

- Apply **Depth-First Search**
  - List the nodes in the order they are visited

- Applications:
  - Scheduling tasks
  - Dependency resolution
  - Course scheduling



# Further questions?