Breadth-First Search

09114319: Data Structures and Algorithms

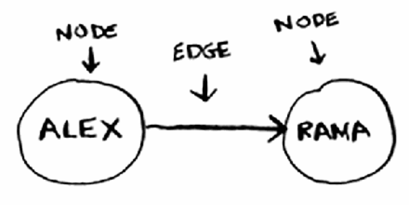
Ratthaprom PROMKAM, Dr. rer. nat.

Department of Mathematics and Computer Science, RMUTT

Outline

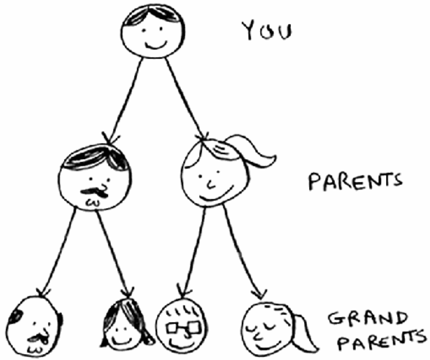
- Introduction to Graphs
- Graph Representation
- Breadth-First Search Algorithm
- Applications of BFS
- Queue Data Structure
- Implementation of BFS
- Performance Analysis

- A graph is a data structure that represents a set of connections.
- It consists of **nodes** (vertices) and **edges** (connections between nodes).



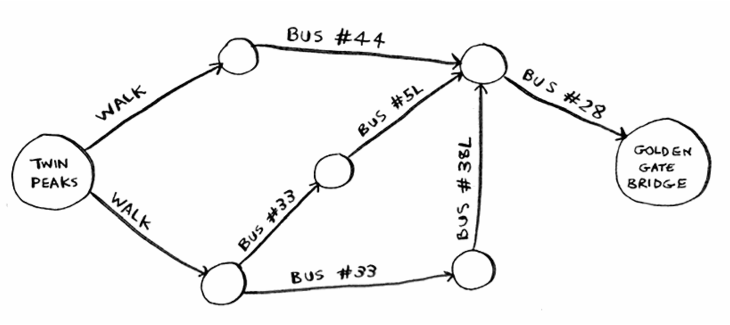
- Examples:
  - Social networks (friends connected to each other)
  - City maps (roads between locations)
  - Game AI (paths between game states)

Introduction to Graphs



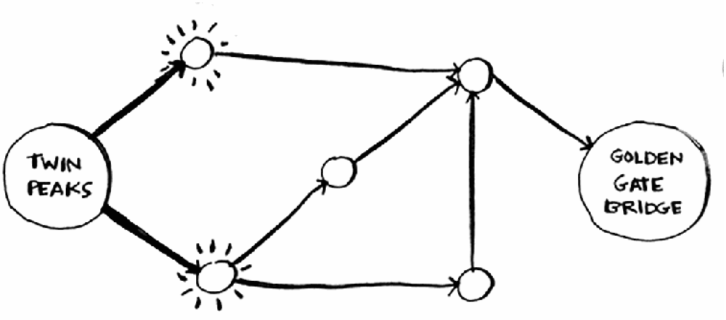
Example: Is there a path?

- Suppose you're in San Francisco, and you want to go from *Twin Peaks* to the *Golden Gate Bridge*.
- You want to get there by bus, with the minimum number of transfers. Here are your options.



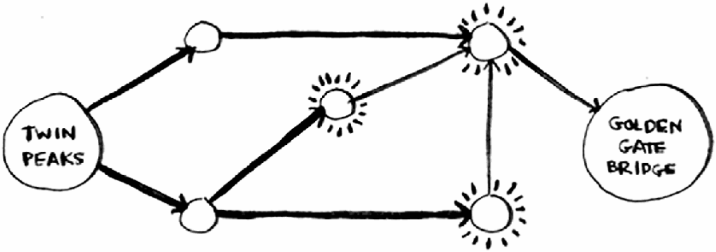
Example: Is there a path?

- Here are all the places you can get to in one step.



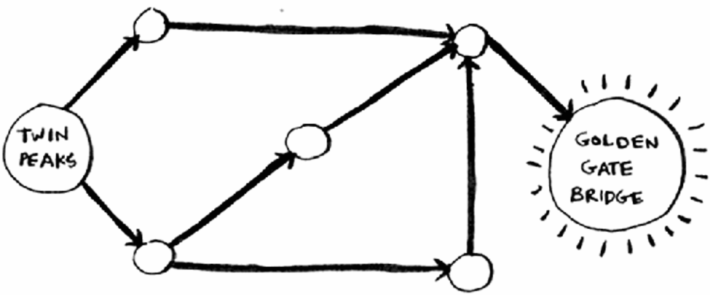
Example: Is there a path?

- The bridge isn't highlighted. You can't get there in one step.
- Can you get there in two steps?



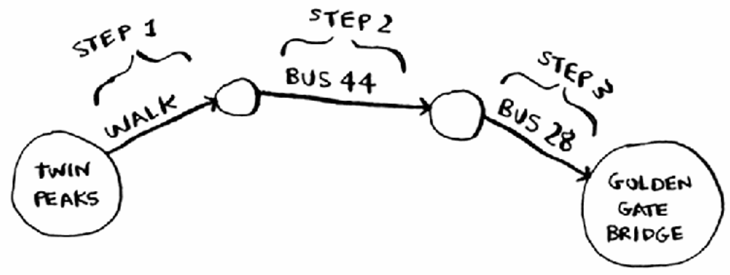
Example: Is there a path?

✎ Again, the bridge isn't there, so you can't get to the bridge in two steps. What about three steps?



Example: Is there a path?

- Aha! Now the Golden Gate Bridge shows up.
- So it takes 3 steps to get from Twin Peaks to the bridge using this route.

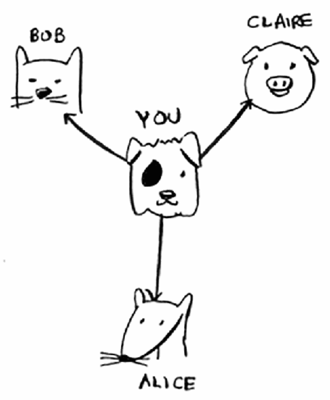


Example: Find Mango Seller on Facebook

- ✎ Suppose you're the proud owner of a mango farm.
- ✎ You're looking for a mango seller who can sell your mangoes.
- ✎ Are you connected to a mango seller on Facebook?

Example: Find Mango Seller on Facebook

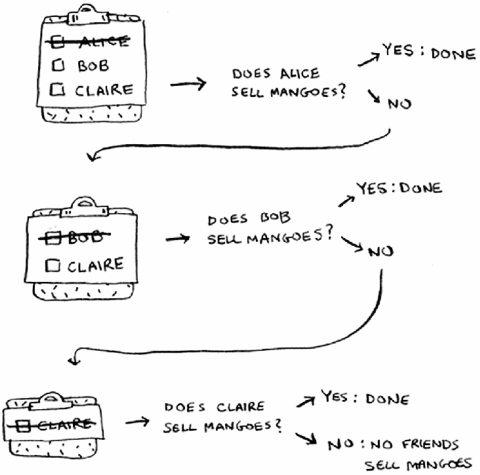
Well, you can search through your friends.



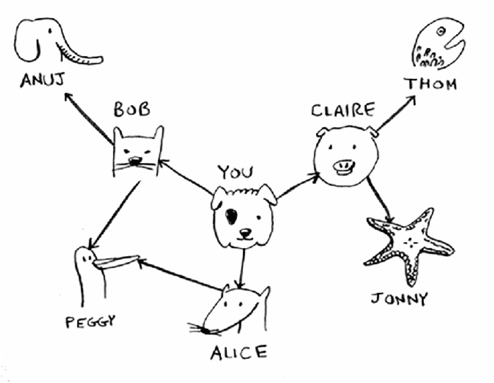
This search is pretty straightforward. First, make a list of friends to search.

Example: Find Mango Seller on Facebook

- ✎ Now, go to each person in the list and check whether that person sells mangoes.

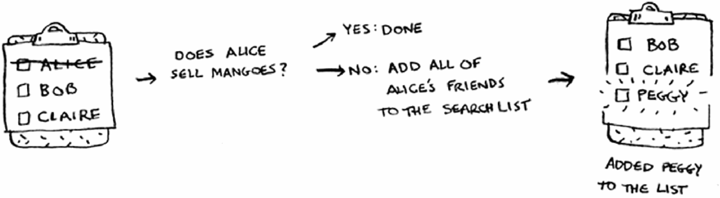


- ✎ Suppose none of your friends are mango sellers. Now you have to search through your friends' friends.



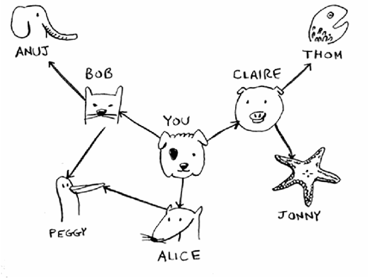
Example: Find Mango Seller on Facebook

- Each time you search for someone from the list, add all of their friends to the list.



- This way, you not only search your friends, but you search their friends, too.
- Remember, the goal is to find one mango seller in your network.
- That means you'll eventually search her friends—and then their friends, and so on.

Graph Representation



Adjacency List: Each node stores a list of its neighbors.

```
graph = {}
graph["you"] = ["alice", "bob", "claire"]
graph["bob"] = ["anuj", "peggy"]
graph["alice"] = ["peggy"]
graph["claire"] = ["thom", "jonny"]
```

Adjacency Matrix: A 2D matrix where rows and columns represent nodes.

```python
import numpy as np

graph = np.array([
    [0, 1, 1, 1, 0, 0, 0, 0],  # you
    [0, 0, 0, 0, 0, 1, 0, 0],  # alice
    [0, 0, 0, 0, 1, 1, 0, 0],  # bob
    [0, 0, 0, 0, 0, 0, 1, 1],  # claire
    [0, 0, 0, 0, 0, 0, 0, 0],  # anuj
    [0, 0, 0, 0, 0, 0, 0, 0],  # peggy
    [0, 0, 0, 0, 0, 0, 0, 0],  # thom
    [0, 0, 0, 0, 0, 0, 0, 0]   # jonny
])
```

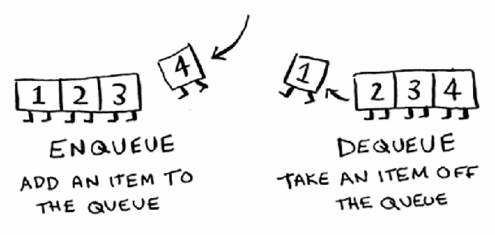
Breadth-First Search (BFS)

- ✎ BFS explores nodes in a **level-order fashion**.
- ✎ It describes a strategy for searching an **unweighted graph**.
- ✎ It uses a **queue** to track nodes to visit.
- ✎ **Steps of BFS:**
  1. Start from a given node.
  2. Explore all its neighbors.
  3. Move to the next level of neighbors.
  4. Repeat until the target node is found or all nodes are explored.

Queue Data Structure

- ✎ BFS uses a queue (FIFO: First In, First Out).

- ✎ Enqueue: Add nodes to the queue.
- ✎ Dequeue: Process and remove nodes from the front.



Example: Queue Operations

Deques are a generalization of stacks and queues (the name is pronounced *deck* and is short for *double-ended queue*).

```python
from collections import deque

queue = deque()
queue.append("Alice")
queue.append("Bob")
print(queue.popleft())  # Alice
print(queue.popleft())  # Bob
```

BFS Algorithm in Python

```python
from collections import deque

def bfs(graph, start, goal):
    queue = deque([start])
    visited = [ ]
    while queue:
        node = queue.popleft()
        if node == goal:
            return True  # Goal found
        if node not in visited:
            visited.append(node)
            neighbors = graph[node]
            queue += neighbors
    return False  # Goal not found
```

Performance Analysis of BFS

- BFS explores each edge once: $O(V + E)$, where:
  - V is the number of vertices (nodes).
  - E is the number of edges (connections).
- Space complexity: $O(V)$ (storing visited nodes and queue).

Recap

- BFS determine a path from the source vertex to the destination vertex in an unweighted graph, if one exists.
- It uses a queue (FIFO) to explore nodes level by level.
- Performance is $O(V + E)$, making it efficient for large graphs.
- Applications include route planning, friend suggestions, and AI decision trees.