Course: Data Structures (CSE CS203A, 114-1)
Study Note: Graph

Student ID: 1131428 Student Name: 徐妙芬

**Definition**

A graph G(V, E) consists of:

1. V = set of vertices (nodes)
2. E = set of edges connecting vertices

**Real-world Motivation: Google Maps**

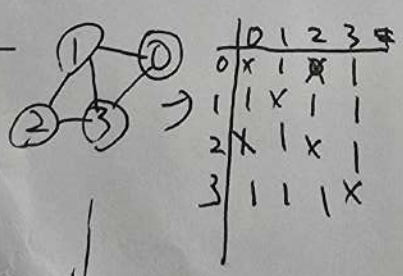
| Concept | Google Maps Analogy |
|---|---|
| Node (Vertex) | Intersection or landmark (e.g., Neili Station, Yuan Ze University) |
| Edge (Link) | Road connecting two intersections |
| Weight | Distance, travel time, or cost |
| Path | Sequence of connected roads from start to destination |
| Graph traversal / search | Finding all reachable locations or the best route |
| Cycle | Round trip that ends at the starting point |
| Directed edge | One-way street |
| Undirected edge | Two-way road |

***Classification***

| *Type* | Description | Example |
|---|---|---|
| *Undirected Graph* | Edges have no direction | Friendship network |
| *Directed Graph (Digraph)* | Edges have direction | Instagram "following" |
| *Weighted Graph* | Each edge has a cost | Google Maps distance |
| *Unweighted Graph* | All edges equal | Board game map |
| *Cyclic Graph* | Has loops | City ring road |
| *Acyclic Graph* | No loops | Family tree |
| *Connected Graph* | Every node reachable | Road network |
| *Disconnected Graph* | Some nodes isolated | Islands without bridge |

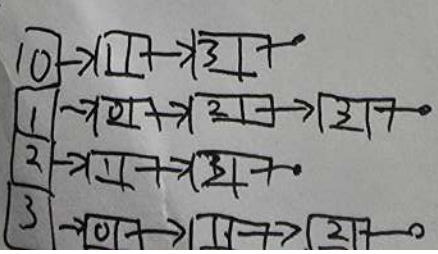
**Graph Representation in Memory**

1. Adjacency Matrix




(array 2D)
=> Pro & Con (same)

2. Adjacency list



(linked list)

2. Abstract

## Graph Traversal

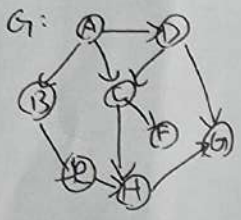
1. Graph vs. Tree

Tree : Hierarchy (階層)

The Same:
have vertase, edage

Difference:
hierarchy , traversal Type · Graph has cycle (if travel , need visit[])

2. Depth-First Search (DFS)




⇒ Stack (LIFO)

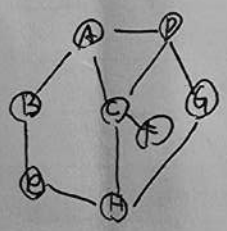
| | visit[] |
|---|---|
| A : B C D | A |
| D : G | AD |
| G : X | ADG |
| C : H F | ADGC |
| F : X | ADGCF |
| H : X | ADGCFH |
| B : E | ADGCFHB |
| E : X | ADGCFHBE |

T : by Level

3. Bread-First Search (BFS)

G : by degree

ex :




⇒ Queue (FIFO)

Q :

| | visit[] |
|---|---|
| A : B C D | A |
| B : E | AB |
| C : H F | ABC |
| D : G | ABCD |
| ⋮ | ⋮ |
| until Q.empty() | ABCDEFG |