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

Student ID: 1111426

Student Name: 廖婧婷

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 ，user可自行定義
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

V×V 的二維陣列表示

→ 代表node

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 0 & 1 \\ 3 & 1 & 0 & 0 & 0 \end{array}$$

→ 1: 有辺,有連接

0: 無連接

$O(|V|^2)$

辺查詢: $O(1)$

適用 : Dense graph

空間效率: 較差, 即使辺少, 仍需$|V|^2$

2. Adjacency List

使用 array 來 store 頂点，array中的元素指向 link list ，像 hash table

$O(|V|+|E|)$ ，辺查詢: 慢, 需看整個 link list

適用 : sparse graph

空間效率: good! 只store real辺

Graph Traversal

1. Graph vs. Tree

The Same: tree 是 graph 的一種
都由 nodes, edges 組成
都可使用 DFS. BFS

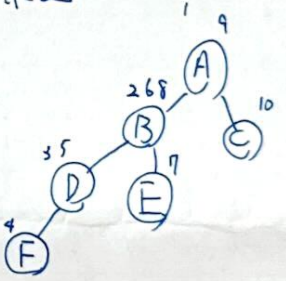
Difference:
tree 不包含 cycles，graph 可多個点 connect，tree 有 root node
边的數量：tree：N個nodes，N-1個边；graph：0～$\frac{N(N-1)}{2}$

2. Depth-First Search (DFS)

用 stack 和 recursion 實現。

1. 從 start node 開始（標記）
2. 訪問這個 nodes 第一個未標記的鄰居
3. 從新鄰居 node 重複②，直到 1. 到達 leave 節点 2. 所有鄰居皆標記
4. Back track 到前一個 node，check 是否有未標記鄰居

※深度優先
應用：path finding，Topological sorting．cycle detection

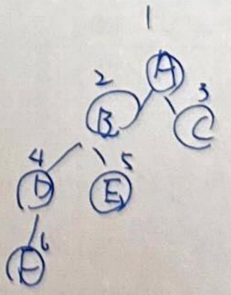


3. Bread-First Search (BFS)

使用 Queue

1. start node 開始，並標記後放入隊列中
2. 從隊列中取出一個 node
3. 訪問這個 node 所有未訪問的鄰居，標記後放入隊列
4. 重複②③

應用：找最短 path，連通分量的識別



by level
去 search