Data Structure : Graph & Tree Basics

Graph

Definition (Graph)

一個圖 (Graph) $G = (V, E)$ 由兩個部分組成：

1. V 是一個有限的集合，稱作 Virtices。
2. E 是一個 V 上的 Binary Relation，稱作 Edge Set。這個 Relation 通常用 $(\_, \_)$ 表示。
3. 如果 E 是 ordered pair，則這個圖稱作「有向圖」；如果 E 是 unordered pair，則稱作無向圖。

Remark

1. 雖然 E 中的元素有 order/unordered 之別，但符號上都用 (u, v) 表示。
2. 在無向圖中 $(u, v) \nRightarrow (u, v)$; 但有向圖中 $(u, v) \Rightarrow (u, v)$

Definition (Incident)

1. 若 $G = (V, E)$ 是有向圖，且 $(u, v) \in E$，則稱 (u, v) 「incident from / leaves u」，且「incident to / enters v」。
2. 若 $G = (V, E)$ 是無向圖，且 $(u, v) \in E$，則稱 (u, v) 「incident on u and v」。

Remark

1. 其實這比較像英文的用語介紹。
2. 接下來的內文為了方便，在有向圖中，用「從 u 離開」表示「incident from/leaves u」；用「進入 u」表示「incident to / enters v」。
3. 用「經過 u 的邊」表示「incident on u」。

Definition (Degree)

1. 若 $G = (V, E)$ 是個有向圖，$u \in V$, 則：

    1. In-degree of u：進入 u 的邊數。

    $$d_{u,i} = |\{(i,u)|(i,u) \in E\}|$$

    2. Out-degree of u：從 u 出發的邊數。

    $$d_{u,o} = |\{(u,i)|(u,i) \in E\}|$$

    3. Degree of u：u 的 in-degree 加 out-degree。

    $$\deg(u) = d_{u,i} + d_{u,o}$$

2. 若 $G = (V, E)$ 是個無向圖，且 $u \in V$，則：

    1. Degree of u：

    $$\deg(u) = |\{(u,i)|(u,i) \in E\}|$$

3. (Hand-Shaking Lemma)
   假定 $G = (V, E)$ 是個無向圖，則：

$$\sum_{u \in V} \deg(u) = 2|E|$$

Definition (Path)

1. 一個從 u 到 u'，長度為 $k \geq 0$ 的 path, p, 定義為一個由 V 中的元素形成的序列：

$$p = \langle v_0, v_1 \dots v_k \rangle$$

且該序列滿足「(v_i, v_{i+1}) 是邊」「頭尾分別是 u, u'」即：

$$\begin{cases} v_0 = u \\ v_k = u' \\ \forall i \in \{0 \dots k-1\}. (v_i, v_{i+1}) \in E \end{cases}$$

因為上面這個東西定義要塞進邏輯裡面實在是太長了，所以用：

$$p \text{ is a path from } \_{1} \text{ to } \_{2}$$

來說明 p 是一個滿足上述性質的序列。

2. (Simple) 若 p is a path from u to u$'$，且 p 的長度為 k。若：

$$\forall (i, j) \in \{(i, j) | 0 \leq i, j \leq k, i \neq j\}. v_i \neq v_j$$

則稱 p 為「simple」 path。因為把 path 的定義跟 simple 的定義全部塞進邏輯裡面實在是太長了，所以就用：

$$p \text{ is a simple path}$$

表示 p 是一個滿足 simple 和 path 性質的序列。

3. (Cycle) 若 p is a path from u to u′，且：

$$u = u'$$

則稱 p 為一個 「cycle」。一樣因為實在是太長了，所以

$$p \text{ is a cycle}$$

表示 p 是一個滿足 cycle 跟 path 性質的序列。

4. (Reachable) 若 (u, v) 之間存在 path, p, 則稱「v is reachable from u via p」。即：

$$\begin{aligned} &\exists k \in \mathbb{N} \cup \{0\}. \exists \langle v_0 \dots v_k \rangle. \\ &\quad v_0 = \_{1}, \\ &\quad v_k = \_{2}, \\ &\quad \forall i \in \{n | 0 \leq n \leq k\}. v_i \in V \end{aligned}$$

因為在 formal logic 中把 Reachable 的定義全部填上去實在太麻煩，所以用：

$$\_{1} \ \_{2} \text{ reachable}$$

表示兩個點 $\_{1}$, $\_{2}$ 是 reachable 的。以及：

$$\_{1} \ \_{2} \text{ is not reachable}$$

表示：$\neg(\_1\ \_2 \text{ reachable})$

另外，因為 k 可以是 0，所以不管是否 (u, u)，u 自己一定跟自己是 reachable 的。

5. (Connected) 假定：

$$\forall \mathrm{u}, \mathrm{u}' \in V. \exists p. p \text{ is a path from u to u}'$$

則稱圖是 Connected. 用：

$$G \text{ is connected}$$

表示 G 是一個滿足 connected 性質的 graph; 用：

$$G \text{ is not connected}$$

表示 $\neg(G \text{ is connected})$

6. (cyclic) 假定：

$$\exists p. p \text{ is a path } \textit{and } p \text{ is a cycle}$$

則稱 G 是 cyclic。相反地，若：

$$\neg(\exists p. p \text{ is a path } \textit{and } p \text{ is a cycle})$$

則稱 G 是 acyclic. 分別用：

$$\begin{cases} G \text{ is cyclic} \\ G \text{ is acyclic} \end{cases}$$

表示有環跟無環。

Thm : 無向圖有 Path ⇒ 有 Simple Path

$$存在 path \Rightarrow 存在\ simple\ path$$

1. $|V| = 2$: 只有兩個點。顯然成立。

2. 假定 $|V| = |V'| - 1$ 時命題成立，則對於任意 $u, v \in V$：
    1. $u, v \in V'$： 由歸納法假設可知命題成立。
    2. $u \in V', v \notin V'$，且 u, v 存在 path，且僅有最後一個點為 v：

        令該 path 為：

        $$p = \langle u, u_1 \dots u_{k-1}, v \rangle$$

        - 因 $v \neq u, u_1 \dots u_{k-1}$ 故 $u, u_1 \dots u_{k-1} \in V'$。
        - 由歸納法假設知 u, u_{k-1} 間存在 simple path, p'。
        - 又 $v \notin V'$，故 $v \notin p'$，因此可知 p 為一 simple path。

    3. $u \in V', v \notin V'$，且 u, v 存在 path，且 v 不只最後一個點為 v：
        - 假定 v 第一次出現在 u_k
        - 取 $p' = \langle u, \dots u_k \rangle$
        - 由於 p' 中 v 僅在最後出現，故套用 2. 可知 $u, u_k = v$ 之間存在 simple path。

Thm：Undirected, Connected ⇒ |E| ≥ |V| - 1

1. |V| = 1 原題顯然成立。

2. 假定 $|V| = |V'| + 1$，隨便挑出一個點 $v \in V$，並令剩下的子圖 $G' = (V', E')$。$V = V' \cup \{v\}$, $v \notin V'$, $E' \subset E$:

    已知：$|E'| \geq |V'| - 1$

    且：$|V| = |V'| + 1$

    若原圖 $G = (V, E) = (V' \cup \{v\}, E)$ 連通，則 $\exists u \in V'. (u, v) =: e \in E$，否則 v 不連通。而 $v \notin V'$，故 $e \notin E'$，因此：

    $$|E| \geq |E'| + 1 \geq |V'| + 1 = |V|$$

3. 由數學歸納法知原題成立。

Free Trees

Definition (樹等價定義們)

下列敘述等價:

1. G 是 Tree $\equiv$ G 是個連通、無環、無向的圖。

$$G \text{ is a tree} \equiv \begin{cases} G \text{ is undirected}, \textit{and} \\ G \text{ is connected}, \ \textit{and} \\ G \text{ is acyclic} \end{cases}$$

2. 任兩點存在唯一 simple path 的圖:

$$\forall u, u' \in V, u \neq u'. \exists! p. \, p \text{ is a simple path}$$

3. 隨便砍一條邊就會不連通的連通圖:

    1. G is connected, and
    2. $\forall e \in E. G' = (V, E \setminus e)$ is not connected

4. 連通, 而且邊數 = 點數 - 1 的圖:

    1. G is connected, and
    2. $|E| = |V| - 1$

5. 無環, 而且邊數 = 點數 - 1 的圖

    1. G is acyclic, and
    2. $|E| = |V| - 1$

6. 無環,

    1. G is acyclic

2. $\forall e \in \{(u_i, u_j) | u_i, u_j \in V, (u_i, u_j) \notin E\}. \forall G' \in \{(V, E \cup e)\}. G'$ is cyclic

證明

1. 「1. ⇒ 2. 」

   反證：

   1. 如果不存在 path，顯然與連通的前提矛盾。
   2. 若存在超過兩個相異 simple path，任選兩條 $p_1 = (a_0 \dots a_{k_a})$，$p_2 = (b_0 \dots b_{k_b})$，其中 $a_0 = b_0$，$a_{k_A} = b_{k_b}$。
      1. 在 p_1, p_2 中，選擇最小的 k_1 與最小的 k_2，使得 $a_{k_1} = b_{k_2}$。以及次小的 k'_1, k'_2，使得 $a_{k'_1} = b_{k'_2}$。
      2. 這樣的 k_1, k_2 與 k'_1, k'_2必定存在，因為最差狀況下 $k_1 = 0, k_2 = 0$，以及 $k'_1 = a_{k_a}, k'_2 = b_{k_b}$。
      3. $p = (a_{k_1} \dots a_{k'_1}, b_{k'_2 - 1} \dots b_{b_{k_1}})$ 為一個 cycle。與前提矛盾。

   Remark : 敘述的意思並不是「任兩點存在的 path 都是 simple path」，而是「如果兩點間有 simple path，則該 simple path 唯一」。

2. 「2. ⇒ 3. 」

   1. 因為任兩點都存在 simple path，故 connected.
   2. 假定 $\exists e = (u, v). G' = (V, E \setminus (u, v))$ is connected，則可知 u, v 仍然 rechable，令這條 path 為 p'，則可知 G 當中，至少有兩個方法構造 u 往 v 的相異 simple path：
      1. p'：因為「u, v 有 path $\Rightarrow u, v$ 有 simple path」
      2. (u, v)

   故：

   $$\neg\,(\exists e = (u, v). G' = (V, E \setminus (u, v)) \text{ is connected})$$

   把 $\neg$ 塞進去之後得證原敘述成立。

3. 「3. ⇒4. 」

前面 Thm 已經證完 $|E| \geq |V| - 1$，僅證 $|E| \leq |V| - 1$ 即可。