

King's College London

5CCS2FC2 Foundations of Computing II

2019 年 1 月考卷 — 逐題詳解

試卷結構說明。本試卷共 13 題，限時兩小時：

- 第 1–10 題（共 50 分）：選擇題，每題有一個或多個正確選項。必須選出所有正確選項且不可多選，選錯會倒扣分數。
- 第 11–13 題（每題 25 分）：三題長答題中任選兩題作答。

本詳解涵蓋全部 13 題：先完整重述並解析每個題目在問什麼、考哪個觀念，再一步一步推導出解答。

Contents

第一部分：選擇題（第 1–10 題）	3
1 遞迴可枚舉（Recursively Enumerable）的定義	[4 分] 3
2 Cook–Levin 定理	[4 分] 3
3 Master 定理：判斷 $\Theta(n \log n)$	[6 分] 4
4 2SAT 蘊涵圖與強連通分量	[4 分] 5
5 2SAT 可滿足性判定	[2 分] 6
6 單體法（Simplex Method）的一次疊代	[7 分] 7
7 旅行推銷員問題與 2-opt 演算法	[6 分] 8
8 r -可近似（ r -approximable）的定義	[4 分] 9
9 相依圖、SCC 與元件圖（Component Graph）	[8 分] 10
10 最小生成樹（Minimum Spanning Tree）	[5 分] 11
第二部分：長答題（第 11–13 題）	13
11 REJECT _{TM} 的不可判定性與遞迴可枚舉性	[25 分] 13
12 數學歸納法、DPLL 演算法、 $\text{SAT} \leq_p \text{3SAT}$	[25 分] 14

13 SAT 歸約到 CLIQUE 與 NP 成員資格

[25 分] 16

第一部分：選擇題（第 1–10 題，共 50 分）

1 遞迴可枚舉（Recursively Enumerable）的定義

[4 分]

題目說明

下列四個敘述中，哪一個正確描述了「判定問題 X 是遞迴可枚舉（recursively enumerable, r.e.）」的意義？（單選）

- A. 存在一個對 X 而言既 sound（可靠）又 complete（完備）的演算法。
- B. X 的每一個演算法都既 sound 又 complete。
- C. 任何對 X 而言 sound 且 complete 的演算法也必定會終止（terminating）。
- D. 不存在對 X 而言 sound、complete 且 terminating 的演算法。

本題考的是本課程中三個核心性質的定義。設演算法 A 針對判定問題 X ：

- **Sound**（可靠）：若 A 回答「是」，則輸入真的屬於 X （不會誤報）。
- **Complete**（完備）：對每一個屬於 X 的輸入， A 都會回答「是」（不會漏報）。
- **Terminating**（終止）： A 對所有輸入（包括不屬於 X 的）都會在有限步內停止。

詳細解答

課程中的標準定義是：

X 可判定（decidable） $\iff$ 存在 sound + complete + terminating 的演算法；

X 遞迴可枚舉（r.e.） $\iff$ 存在 sound + complete 的演算法（不要求終止）。

換言之，r.e. 問題擁有一個「半判定程序」（semi-decision procedure）：對 yes-instance 一定會在有限時間內回答「是」；但對 no-instance 可能永遠跑不完。

逐一檢視選項：

- **A.** 「存在一個 sound 且 complete 的演算法」——這正是 r.e. 的定義。正確。
- **B.** 「每一個演算法都 sound 且 complete」——顯然錯誤：對任何問題我們都能寫出永遠回答「否」的演算法（不 complete）。定義要求的是存在，不是全部。
- **C.** 若 sound + complete 就必定 terminating，那 r.e. 就等於 decidable。但停機問題（Halting Problem）是 r.e. 卻不可判定，故錯誤。
- **D.** 這是描述「 X 不可判定」，而非「 X 是 r.e.」。兩者是不同性質（且 r.e. 的問題也可能是可判定的），故錯誤。

正確答案

A

2 Cook–Levin 定理

[4 分]

題目說明

下列哪一個敘述正確描述了 Cook–Levin 定理？（單選）

- A. 每一個圖靈機問題都可以多項式時間歸約成 SAT 的實例。
- B. 每一個可以在多項式時間內驗證的問題，都可以在多項式時間內求解。
- C. 圖靈機的停機問題是不可判定的。
- D. 布林可滿足性問題 SAT 是 NP-complete。

詳細解答

Cook-Levin 定理 (1971/1973) 的內容是：**SAT** 是 **NP-complete**。也就是說：

1. $\text{SAT} \in \text{NP}$ (給定一組真值指派，可在多項式時間驗證它是否滿足公式)；
2. 對每一個 $Y \in \text{NP}$ ，都有 $Y \leq_p \text{SAT}$ (NP-hard)。

它是歷史上第一個被證明為 NP-complete 的問題，也是後續所有 NP-completeness 歸約鏈的起點。

逐一檢視選項：

- **A.** 錯誤。可歸約到 SAT 的是「NP 中的問題」，而不是「每一個圖靈機問題」。例如停機問題不在 NP 中，無法多項式歸約到 SAT (否則 SAT 就不可判定了)。
- **B.** 這其實是「 $P = \text{NP}$ 」的敘述，是著名的未解問題，並非 Cook-Levin 定理。錯誤。
- **C.** 這是 Turing (1936) 的停機問題不可判定性定理，與 Cook-Levin 無關。錯誤。
- **D.** 正是 Cook-Levin 定理的內容。正確。

正確答案

D

3 Master 定理：判斷 $\Theta(n \log n)$

[6 分]

題目說明

利用 Master 定理，判斷下列哪些遞迴關係式所描述的函數成長率為 $\Theta(n \log n)$ (皆假設 $T(1) = 1$)。(複選)

- A. $T(n) = 4T(n/2) + n \log n$
- B. $T(n) = T(n/4) + n \log n$
- C. $T(n) = 5T(n/5) + n$
- D. $T(n) = 5T(n/2) + n^3$

觀念補充

Master 定理。 對 $T(n) = aT(n/b) + f(n)$ ($a \geq 1, b > 1$)，令臨界指數 $c^* = \log_b a$ ：

- **Case 1**：若 $f(n) = O(n^{c^*-\epsilon})$ ($\epsilon > 0$)，則 $T(n) = \Theta(n^{c^*})$ 。
- **Case 2**：若 $f(n) = \Theta(n^{c^*})$ ，則 $T(n) = \Theta(n^{c^*} \log n)$ 。
- **Case 3**：若 $f(n) = \Omega(n^{c^*+\epsilon})$ 且滿足正則條件 $af(n/b) \leq cf(n)$ (某 $c < 1$)，則 $T(n) = \Theta(f(n))$ 。

詳細解答

選項 **A**： $a = 4, b = 2, c^* = \log_2 4 = 2$ 。 $f(n) = n \log n$ 。由於 $n \log n = O(n^{2-\epsilon})$ (取 $\epsilon = 0.5$ 即可，因為 $\log n$ 的成長慢於任何正冪次)，屬於 **Case 1**，故

$$T(n) = \Theta(n^2) \neq \Theta(n \log n). \quad \text{不符。}$$

選項 **B**： $a = 1, b = 4, c^* = \log_4 1 = 0$ ，即 $n^{c^*} = 1$ 。 $f(n) = n \log n = \Omega(n^{0+\epsilon})$ (取 $\epsilon = 1$)，檢查正則條件：

$$af(n/b) = \frac{n}{4} \log \frac{n}{4} \leq \frac{1}{4} n \log n = cf(n), \quad c = \frac{1}{4} < 1. \checkmark$$

屬於 **Case 3**，故 $T(n) = \Theta(f(n)) = \Theta(n \log n)$ 。符合。

選項 C： $a = 5, b = 5, c^* = \log_5 5 = 1$ 。 $f(n) = n = \Theta(n^1)$ ，恰好等於臨界冪次，屬於 Case 2，故

$$T(n) = \Theta(n^1 \log n) = \Theta(n \log n). \quad \text{符合。}$$

(這正是 merge sort 型遞迴的一般化：5 路分割、線性合併。)

選項 D： $a = 5, b = 2, c^* = \log_2 5 \approx 2.32$ 。 $f(n) = n^3 = \Omega(n^{2.32+\epsilon})$ ，檢查正則條件：

$$a f(n/b) = 5 \left(\frac{n}{2}\right)^3 = \frac{5}{8} n^3 \leq c n^3, \quad c = \frac{5}{8} < 1. \checkmark$$

屬於 Case 3，故 $T(n) = \Theta(n^3) \neq \Theta(n \log n)$ 。不符。

正確答案

B、C

4 2SAT 蘊涵圖與強連通分量

[4 分]

題目說明

考慮以下 2SAT 實例：

$$(\neg Q \vee P) \wedge (\neg P \vee \neg S) \wedge (\neg P \vee \neg R) \wedge (S \vee R) \wedge (Q \vee R)$$

畫出蘊涵圖 (implication graph) 並找出強連通分量 (SCC)。下列哪些子集是蘊涵圖的強連通分量？(複選)

- A. $\{P, Q, \neg R, S\}$ B. $\{\neg S\}$ C. $\{\neg P, \neg R, \neg S\}$ D. $\{\neg P, Q, S\}$ E. $\{\neg P, \neg Q, R\}$

觀念補充

蘊涵圖的建構。每個子句 $(a \vee b)$ 邏輯上等價於兩條蘊涵：

$$\neg a \rightarrow b \quad \text{以及} \quad \neg b \rightarrow a.$$

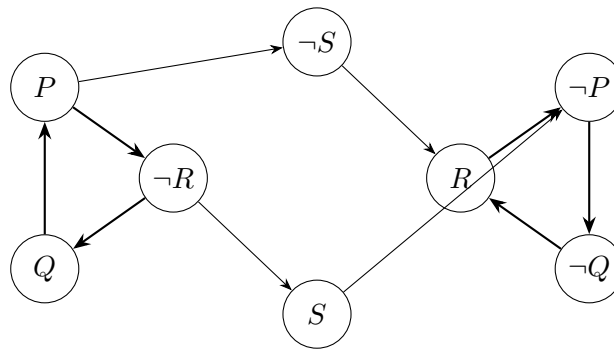
蘊涵圖以全部 $2n$ 個文字 (每個變數及其否定) 為節點，為每個子句加入上述兩條有向邊。

詳細解答

步驟一：由每個子句寫出兩條蘊涵邊。

$$\begin{aligned} (\neg Q \vee P) : & \quad Q \rightarrow P, \quad \neg P \rightarrow \neg Q \\ (\neg P \vee \neg S) : & \quad P \rightarrow \neg S, \quad S \rightarrow \neg P \\ (\neg P \vee \neg R) : & \quad P \rightarrow \neg R, \quad R \rightarrow \neg P \\ (S \vee R) : & \quad \neg S \rightarrow R, \quad \neg R \rightarrow S \\ (Q \vee R) : & \quad \neg Q \rightarrow R, \quad \neg R \rightarrow Q \end{aligned}$$

步驟二：畫出蘊涵圖。(節點為 8 個文字，邊為上述 10 條)



步驟三：找出強連通分量。(SCC = 任兩點之間互相可達的最大節點集合)

- 迴圈 $P \rightarrow \neg R \rightarrow Q \rightarrow P$ 使 $\{P, Q, \neg R\}$ 互相可達，構成一個 SCC。
- 迴圈 $\neg P \rightarrow \neg Q \rightarrow R \rightarrow \neg P$ 使 $\{\neg P, \neg Q, R\}$ 構成一個 SCC (這正是前者的「鏡像」——2SAT 蘊涵圖天生具有對偶對稱性)。
- S 的唯一出邊是 $S \rightarrow \neg P$ ，之後只能在 $\{\neg P, \neg Q, R\}$ 中繞圈，永遠回不到 S ，故 $\{S\}$ 自成一個單點 SCC。
- 同理 $\neg S$ 的唯一出邊是 $\neg S \rightarrow R$ ，回不到 $\neg S$ ，故 $\{\neg S\}$ 也自成一個單點 SCC。

所以四個 SCC 為：

$$\{P, Q, \neg R\}, \quad \{\neg P, \neg Q, R\}, \quad \{S\}, \quad \{\neg S\}.$$

步驟四：檢查各選項。

- A. $\{P, Q, \neg R, S\}$ ： S 不屬於 $\{P, Q, \neg R\}$ 這個 SCC (從 S 出發到不了 P)，錯誤。
- B. $\{\neg S\}$ ：單點 SCC，正確。
- C. $\{\neg P, \neg R, \neg S\}$ ： $\neg R$ 與 $\neg P$ 分屬不同 SCC，錯誤。
- D. $\{\neg P, Q, S\}$ ：三者分屬三個不同 SCC，錯誤。
- E. $\{\neg P, \neg Q, R\}$ ：正確。

正確答案

B、E

5 2SAT 可滿足性判定

[2 分]

題目說明

利用上一題建構的蘊涵圖，判斷第 4 題的公式是否可滿足。

A. Satisfiable (可滿足) B. Unsatisfiable (不可滿足)

觀念補充

2SAT 判準 (Aspvall–Plass–Tarjan)。2SAT 公式不可滿足 $\iff$ 存在某變數 X ，使 X 與 $\neg X$ 落在同一個強連通分量中。

詳細解答

由第 4 題，四個 SCC 為 $\{P, Q, \neg R\}$ 、 $\{\neg P, \neg Q, R\}$ 、 $\{S\}$ 、 $\{\neg S\}$ 。逐一檢查每個變數：

- P 與 $\neg P$ ：分屬第一、第二個 SCC。✓

- Q 與 $\neg Q$ ：分屬第一、第二個 SCC。✓
- R 與 $\neg R$ ：分屬第二、第一個 SCC。✓
- S 與 $\neg S$ ：各自是單點 SCC。✓

沒有任何變數與其否定同處一個 SCC，故公式可滿足。

驗證（實際找一組解）：取 $P = \text{False}$, $Q = \text{False}$, $R = \text{True}$, $S = \text{False}$ ：

$$(\neg Q \vee P) = \top, \quad (\neg P \vee \neg S) = \top, \quad (\neg P \vee \neg R) = \top, \quad (S \vee R) = \top, \quad (Q \vee R) = \top. \checkmark$$

正確答案

A (Satisfiable)

6 單體法 (Simplex Method) 的一次疊代

[7 分]

題目說明

考慮以下初始單體表 (tableau)：

$$T_1 = \begin{array}{c|cccccc|c} & x & y & s_1 & s_2 & s_3 & C & \\ \hline -9 & -5 & 1 & 0 & 0 & 0 & 9 \\ 2 & -6 & 0 & 1 & 0 & 0 & 8 \\ -3 & 10 & 0 & 0 & 1 & 0 & 7 \\ -9 & -7 & 0 & 0 & 0 & 1 & 0 \end{array}$$

找出樞軸 (pivot)，對 T_1 執行一次單體法疊代得到第二個表 T_2 。依照題目指定的位置，求 T_2 中的 X (第 3 列 x 行)、 Y (第 4 列 y 行)、 Z (第 1 列 s_2 行)、 W (第 2 列最右行) 之值。(單選)

- A. $X = -0.3$, $Y = 0$, $Z = 0$, $W = 9.8$
- B. $X = 0$, $Y = -34$, $Z = 4.5$, $W = 4$
- C. $X = 0$, $Y = -2$, $Z = 0$, $W = 6$
- D. $X = 0$, $Y = 0$, $Z = 0$, $W = 1$

詳細解答

步驟一：選擇樞軸行 (entering variable)。看目標函數列 (最後一列) $(-9, -7, 0, 0, 0)$ ，選最負的係數： x 行的 -9 比 y 行的 -7 更負，故選 x 行為進入變數。

步驟二：比值測試 (ratio test) 選樞軸列。只對樞軸行中嚴格為正的元素計算「右端值 $\div$ 該元素」：

列	x 行元素	比值
1	$-9 (\leq 0)$	不考慮
2	$2 (> 0)$	$8/2 = 4$
3	$-3 (\leq 0)$	不考慮

唯一候選是第 2 列，故樞軸元素為第 2 列、 x 行的 2。

步驟三：樞軸列正規化。第 2 列除以 2：

$$R'_2 = (1, -3, 0, 0.5, 0, 0 \mid 4).$$

步驟四：消去其他列的 x 行。

$$R'_1 = R_1 + 9R'_2 = (-9 + 9, -5 - 27, 1, 0 + 4.5, 0, 0 \mid 9 + 36) = (0, -32, 1, 4.5, 0, 0 \mid 45),$$

$$R'_3 = R_3 + 3R'_2 = (-3 + 3, 10 - 9, 0, 1.5, 1, 0 \mid 7 + 12) = (0, 1, 0, 1.5, 1, 0 \mid 19),$$

$$R'_4 = R_4 + 9R'_2 = (-9 + 9, -7 - 27, 0, 4.5, 0, 1 \mid 0 + 36) = (0, -34, 0, 4.5, 0, 1 \mid 36).$$

步驟五：寫出 T_2 並讀取指定位置。

$$T_2 =$$

x	y	s_1	s_2	s_3	C	
0	-32	1	4.5	0	0	45
1	-3	0	0.5	0	0	4
0	1	0	1.5	1	0	19
0	-34	0	4.5	0	1	36

對照題目給的位置模板： Z 在第 1 列 s_2 行 = 4.5； W 在第 2 列右端 = 4； X 在第 3 列 x 行 = 0； Y 在第 4 列 y 行 = -34。

正確答案

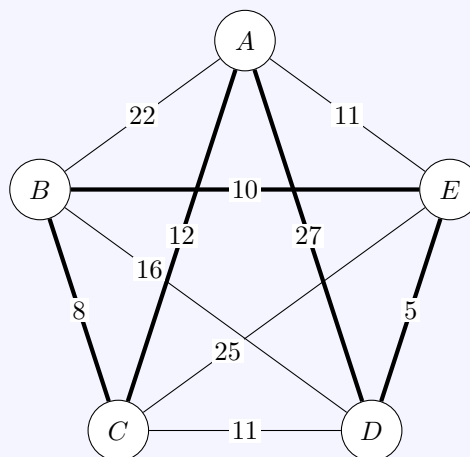
B ($X = 0, Y = -34, Z = 4.5, W = 4$)

7 旅行推銷員問題與 2-opt 演算法

[6 分]

題目說明

考慮以下 TSP 實例 (K_5 完全圖，粗線為目前路線)：



粗體標示的路線為 $A \rightarrow C \rightarrow B \rightarrow E \rightarrow D \rightarrow A$ ，長度 62。使用 2-opt 演算法，下列哪些修改是能得到更短路線的有效交換？(複選)

- A. 移除 $(A, D), (C, B)$ ，加入 $(C, D), (A, B)$ 。
- B. 移除 $(A, C), (B, E)$ ，加入 $(A, B), (C, E)$ 。
- C. 移除 $(A, C), (A, D)$ ，加入 $(C, D), (A, A)$ 。
- D. 移除 $(A, D), (B, E)$ ，加入 $(A, B), (D, E)$ 。
- E. 移除 $(A, D), (B, E)$ ，加入 $(A, E), (B, D)$ 。

觀念補充

2-opt 交換。從目前的迴路中移除兩條不相鄰的邊，迴路斷成兩段路徑；再用唯一另一種方式把兩段的端點重新接起來（等價於把其中一段反轉）。合法的 2-opt 交換必須產生一條單一的新迴路。

詳細解答

邊的權重整理：

	A	B	C	D	E
A	—	22	12	27	11
B	22	—	8	16	10
C	12	8	—	11	25
D	27	16	11	—	5
E	11	10	25	5	—

目前路線 $A \rightarrow C \rightarrow B \rightarrow E \rightarrow D \rightarrow A$ 的邊為 $(A, C), (C, B), (B, E), (E, D), (D, A)$ ，總長 $12 + 8 + 10 + 5 + 27 = 62$ 。

選項 A：移除 $(A, D), (C, B)$ ，加入 $(C, D), (A, B)$ 。移除後迴路斷成兩段： $A-C$ 與 $B-E-D$ 。加入 (C, D) 與 (A, B) 把端點 $\{A, C\}$ 與 $\{B, D\}$ 交叉重接，得到單一迴路

$$A \rightarrow C \rightarrow D \rightarrow E \rightarrow B \rightarrow A, \quad \text{長度} = 12 + 11 + 5 + 10 + 22 = 60 < 62.$$

是合法交換且更短。正確。

選項 B：移除 $(A, C), (B, E)$ ，加入 $(A, B), (C, E)$ 。移除後兩段為 $C-B$ 與 $E-D-A$ 。加入 $(A, B), (C, E)$ 得到單一迴路

$$A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A, \quad \text{長度} = 22 + 8 + 25 + 5 + 27 = 87 > 62.$$

交換合法，但路線變長，不符合「得到更短路線」的要求。錯誤。

選項 C：移除 $(A, C), (A, D)$ ，加入 $(C, D), (A, A)$ 。兩條被移除的邊在頂點 A 相鄰，且 (A, A) 是自環 (self-loop)，根本不是合法的圖邊；結果也不是一條經過五個城市的迴路。無效交換，錯誤。

選項 D：移除 $(A, D), (B, E)$ ，加入 $(A, B), (D, E)$ 。移除後兩段為 $A-C-B$ (端點 A, B) 與 $E-D$ (端點 E, D)。加入 (A, B) 是把同一段的兩端接起來，會形成小圈 $A \rightarrow C \rightarrow B \rightarrow A$ ；加入 (D, E) 則重複了尚在迴路中的邊 (E, D) ，形成另一個二頂點小圈。結果是兩個不相交的迴圈，不是一條完整路線。無效交換，錯誤。

選項 E：移除 $(A, D), (B, E)$ ，加入 $(A, E), (B, D)$ 。兩段同上 ($A-C-B$ 與 $E-D$)，加入 $(A, E), (B, D)$ 交叉重接兩段端點，得到單一迴路

$$A \rightarrow C \rightarrow B \rightarrow D \rightarrow E \rightarrow A, \quad \text{長度} = 12 + 8 + 16 + 5 + 11 = 52 < 62.$$

是合法交換且更短。正確。

正確答案

A、E

8 r -可近似 (r -approximable) 的定義

[4 分]

題目說明

「最佳化問題 X 是 r -approximable ($r \geq 1$)」是什麼意思？(單選)

- A. X 不存在多項式時間的精確演算法。
- B. 存在一個多項式時間演算法，在至多 r 次疊代後回傳最佳解。
- C. X 的每一個多項式時間近似演算法之近似比都 $\geq r$ 。
- D. 存在一個多項式時間演算法，回傳的近似解不會比全域最佳解差超過 r 倍。

詳細解答

定義：最佳化問題 X 是 r -approximable，若存在多項式時間演算法 $\mathcal{A}$ ，對每個輸入都回傳一個可行解，且其值 $\mathcal{A}(I)$ 滿足（以最小化問題為例）

$$\text{OPT}(I) \leq \mathcal{A}(I) \leq r \cdot \text{OPT}(I).$$

也就是「保證品質不劣於最佳解的 r 倍」。例如：具三角不等式的 TSP 有 2-近似演算法（MST 加倍法）、Vertex Cover 有 2-近似演算法。

逐一檢視選項：

- A. 這是在描述「 X 很難精確求解」，與近似保證無關；而且若 X 本身可在多項式時間精確求解，它自動是 1-approximable，兩者並不衝突。錯誤。
- B. 「 r 次疊代後回傳最佳解」把 r 誤解為疊代次數； r 是解的品質比，不是次數。錯誤。
- C. 方向完全相反：這是說「做不到比 r 更好」（不可近似性下界），而 r -approximable 是說「至少能做到 r 」（上界）。錯誤。
- D. 正是定義。正確。

正確答案

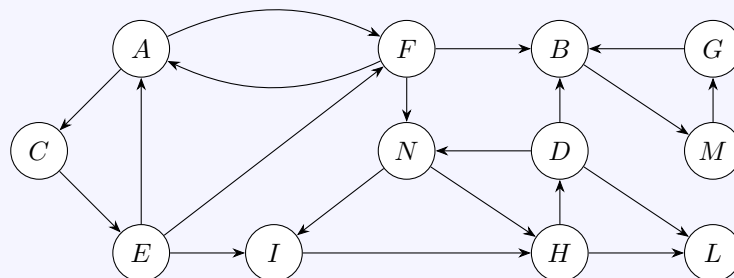
D

9 相依圖、SCC 與元件圖（Component Graph）

[8 分]

題目說明

考慮以下相依圖（dependency graph）：



先找出所有強連通分量並建構元件圖（component graph）。下列哪些邊可以安全地加入相依圖，而不會使強連通分量的數目減少？（複選）

- A. 加入 (L, A) 。 B. 加入 (M, L) 。 C. 加入 (C, F) 。 D. 加入 (M, D) 。

詳細解答

步驟一：找出所有 SCC。沿著有向迴圈找互相可達的節點集合：

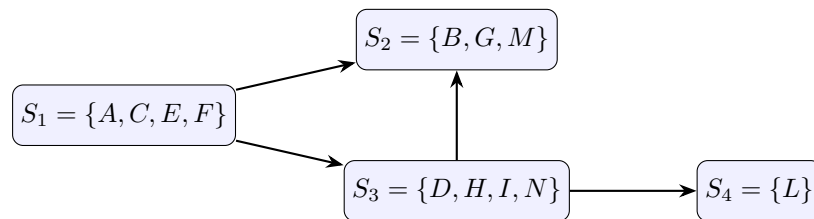
- 迴圈 $A \rightarrow C \rightarrow E \rightarrow A$ 及 $A \leftrightarrow F$ ($E \rightarrow F$ 亦然)： $\{A, C, E, F\}$ 互相可達，記為 S_1 。
- 迴圈 $B \rightarrow M \rightarrow G \rightarrow B$ ： $\{B, G, M\}$ 互相可達，記為 S_2 。
- 迴圈 $D \rightarrow N \rightarrow I \rightarrow H \rightarrow D$ ($N \rightarrow H$ 亦在其中)： $\{D, H, I, N\}$ 互相可達，記為 S_3 。
- L 只有入邊 ($H \rightarrow L$ 、 $D \rightarrow L$)、沒有出邊，自成單點 SCC，記為 $S_4 = \{L\}$ 。

共 4 個 SCC。

步驟二：建構元件圖。把每個 SCC 縮成一點，保留跨 SCC 的邊：

- $F \rightarrow B$ 給出 $S_1 \rightarrow S_2$ ；
- $F \rightarrow N$ 、 $E \rightarrow I$ 給出 $S_1 \rightarrow S_3$ ；

- $D \rightarrow B$ 給出 $S_3 \rightarrow S_2$;
- $H \rightarrow L$ 、 $D \rightarrow L$ 給出 $S_3 \rightarrow S_4$ 。



步驟三：判斷加邊是否安全。關鍵原理：

加入邊 (u, v) 會合併 $SCC \iff$ 元件圖中「 v 所在的 SCC 」已能到達「 u 所在的 SCC 」
(因為這樣會在元件圖中形成迴圈，把路徑上所有 SCC 併成一個，使 SCC 數目減少)。

- **A.** 加入 (L, A) : $L \in S_4$ 、 $A \in S_1$ 。元件圖中 $S_1 \rightarrow S_3 \rightarrow S_4$ ，即 A 所在的 S_1 可達 L 所在的 S_4 ；加入 $L \rightarrow A$ 會形成迴圈 $S_1 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1$ ，把 S_1, S_3, S_4 合併成一個 SCC (4 個變 2 個)。不安全。
- **B.** 加入 (M, L) : $M \in S_2$ 、 $L \in S_4$ 。 $S_4 = \{L\}$ 沒有任何出邊，不可能到達 S_2 ；加入後元件圖只是多了一條邊 $S_2 \rightarrow S_4$ ，仍是 DAG， SCC 數目維持 4。安全，正確。
- **C.** 加入 (C, F) : C 與 F 本來就同屬 S_1 。在同一個 SCC 內部加邊完全不改變 SCC 的劃分，數目仍是 4。安全，正確。
- **D.** 加入 (M, D) : $M \in S_2$ 、 $D \in S_3$ 。元件圖中已有 $S_3 \rightarrow S_2$ (來自 $D \rightarrow B$)；加入 $M \rightarrow D$ 會形成迴圈 $S_2 \rightarrow S_3 \rightarrow S_2$ ，把 S_2 與 S_3 合併 (4 個變 3 個)。不安全。

正確答案

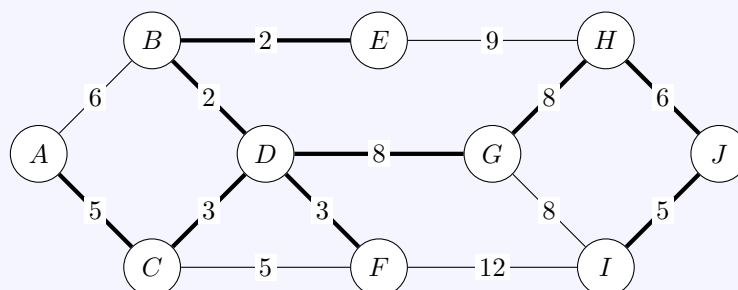
B、C

10 最小生成樹 (Minimum Spanning Tree)

[5 分]

題目說明

考慮以下含十個頂點的加權圖 G (粗線為下面解答求出的 MST 邊)：



使用 Kruskal 或 Prim 演算法建構 G 的最小生成樹。最小生成樹的權重是多少？(單選)

- A. 37 B. 39 C. 42 D. 44

詳細解答

全部 14 條邊： $AB=6$, $AC=5$, $BE=2$, $BD=2$, $CD=3$, $CF=5$, $DF=3$, $DG=8$, $EH=9$, $GH=8$, $GI=8$, $HJ=6$

使用 **Kruskal** 演算法：把邊依權重由小到大排序，依序嘗試加入；若會形成迴圈就捨棄，直到選滿 $n - 1 = 9$ 條邊。

權重	邊	決定	理由 / 目前的連通分量
2	$B-E$	加入	$\{B, E\}$
2	$B-D$	加入	$\{B, D, E\}$
3	$C-D$	加入	$\{B, C, D, E\}$
3	$D-F$	加入	$\{B, C, D, E, F\}$
5	$A-C$	加入	$\{A, B, C, D, E, F\}$
5	$C-F$	捨棄	C, F 已連通 (經 $C-D-F$)，會成迴圈
5	$I-J$	加入	$\{I, J\}$ (另一分量)
6	$A-B$	捨棄	A, B 已連通，會成迴圈
6	$H-J$	加入	$\{H, I, J\}$
8	$D-G$	加入	併入 $G: \{A, \dots, G\}$
8	$G-H$	加入	兩大分量合併，全部 10 個頂點連通 ✓

已選滿 9 條邊，演算法停止 (其後的 $GI=8$ 、 $EH=9$ 、 $FI=12$ 均不需考慮)。

MST 總權重：

$$2 + 2 + 3 + 3 + 5 + 5 + 6 + 8 + 8 = 42.$$

(檢驗：9 條邊、10 個頂點、無迴圈且連通，確為生成樹；Kruskal 的貪婪選擇保證其為最小。)

正確答案

C (權重 = 42)

第二部分：長答題（第 11–13 題，任選兩題，每題 25 分）

11 REJECT_{TM} 的不可判定性與遞迴可枚舉性

[25 分]

題目說明

定義語言

$\text{REJECT}_{TM} = \{\langle M, w \rangle : M \text{ 編碼一台會拒絕 (reject) 輸入 } w \text{ 的圖靈機}\}.$

- [13 分] 從第一原理（即不借助其他已知不可判定問題的歸約）證明 REJECT_{TM} 不可判定。
- [7 分] 建構一個 sound 且 complete 的演算法來辨識所有 $\langle M, w \rangle \in \text{REJECT}_{TM}$ ，證明 REJECT_{TM} 是遞迴可枚舉的。
- [5 分] 據此（或以其他方式）證明補語言 $\overline{\text{REJECT}_{TM}}$ 不是遞迴可枚舉的。

(a) 不可判定性：對角線法（13 分）

「從第一原理」意指要仿照停機問題的經典證明，使用對角線論證（**diagonalisation**）自己導出矛盾。

假設（為了導出矛盾）：存在一台總是停機的圖靈機 R 判定 REJECT_{TM} ，即對任意輸入 $\langle M, w \rangle$ ：

$$R(\langle M, w \rangle) = \begin{cases} \text{accept}, & \text{若 } M \text{ 拒絕 } w, \\ \text{reject}, & \text{若 } M \text{ 不拒絕 } w \text{ (接受或不停機)}, \end{cases}$$

且 R 對所有輸入都會停止。

建構對角線機器 D ：定義新的圖靈機 D ，輸入為某台圖靈機的編碼 $\langle M \rangle$ ：

- 模擬執行 R 於輸入 $\langle M, \langle M \rangle \rangle$ （把 M 自己的編碼餵給 M 當輸入來詢問 R ）；
- 若 R 接受（表示「 M 拒絕 $\langle M \rangle$ 」），則 D 接受；
- 若 R 拒絕（表示「 M 不拒絕 $\langle M \rangle$ 」），則 D 拒絕。

因為 R 總是停機， D 也對所有輸入停機，且必定以「接受」或「拒絕」兩者之一結束。

對角化：把 D 餵給自己。考慮 D 在輸入 $\langle D \rangle$ 上的行為，只有兩種可能：

- 情況一： D 接受 $\langle D \rangle$ 。依 D 的定義，這發生的條件是 R 接受 $\langle D, \langle D \rangle \rangle$ ，也就是「 D 拒絕 $\langle D \rangle$ 」。於是 D 同時接受又拒絕 $\langle D \rangle$ ——一台機器在同一輸入上只能進入一種停機狀態，矛盾。
- 情況二： D 拒絕 $\langle D \rangle$ 。依 D 的定義，這發生的條件是 R 拒絕 $\langle D, \langle D \rangle \rangle$ ，即「 D 不拒絕 $\langle D \rangle$ 」。但我們正處於「 D 拒絕 $\langle D \rangle$ 」的情況，矛盾。

兩種情況都導致矛盾，而 D 又必定停在其中一種狀態，故假設錯誤：判定 REJECT_{TM} 的總是停機的機器 R 不存在。

∴ REJECT_{TM} 不可判定。 ■

(b) 遞迴可枚舉性：建構半判定演算法（7 分）

要證明 REJECT_{TM} 是 r.e.，只需給出一個 sound 且 complete（但不必終止）的演算法。利用通用圖靈機（**Universal TM**）可模擬任何被編碼的機器：

演算法 $\mathcal{S}$ ：輸入 $\langle M, w \rangle$ 。

1. 用通用圖靈機一步一步模擬 M 在輸入 w 上的執行；
2. 若模擬中 M 停在拒絕狀態： $\mathcal{S}$ 回答「接受」；
3. 若模擬中 M 停在接受狀態： $\mathcal{S}$ 回答「拒絕」（或進入無窮迴圈皆可）；
4. 若 M 永不停機： $\mathcal{S}$ 的模擬也永遠進行下去（不回答）。

Soundness（可靠性）： $\mathcal{S}$ 只有在模擬中親眼看到 M 進入拒絕狀態時才回答「接受」；此時 M 確實拒絕 w ，故 $\langle M, w \rangle \in \text{REJECT}_{TM}$ 。 $\mathcal{S}$ 絕不誤報。✓

Completeness（完備性）：若 $\langle M, w \rangle \in \text{REJECT}_{TM}$ ，則 M 在 w 上經過有限多步後停在拒絕狀態；忠實的逐步模擬也會在同樣有限步後看到這件事，於是 $\mathcal{S}$ 必定回答「接受」。 $\mathcal{S}$ 絕不漏報。✓

$\mathcal{S}$ 既 sound 又 complete，故 REJECT_{TM} 是遞迴可枚舉的。■

（注意 $\mathcal{S}$ 不 terminating：當 M 在 w 上不停機時， $\mathcal{S}$ 也永遠不停。這正是它只能「半判定」的原因，與 (a) 的不可判定性一致。）

(c) 補語言不是遞迴可枚舉的（5 分）

觀念補充

關鍵定理。若語言 X 與其補 $\bar{X}$ 都是遞迴可枚舉的，則 X 可判定。

證明概要：設 $\mathcal{S}_1$ 、 $\mathcal{S}_2$ 分別是 X 、 $\bar{X}$ 的半判定演算法。給定輸入 x ，以「交錯執行」(dovetailing) 並行模擬 $\mathcal{S}_1(x)$ 與 $\mathcal{S}_2(x)$ ——輪流各執行一步。因為 x 必屬於 X 或 $\bar{X}$ 其中之一，由完備性，兩者中必有一個會在有限步內接受；哪個先接受就據以回答「 $x \in X$ 」或「 $x \notin X$ 」。此程序 sound、complete 且對所有輸入終止，故 X 可判定。

套用到本題：反設 $\overline{\text{REJECT}_{TM}}$ 是遞迴可枚舉的。由 (b)， REJECT_{TM} 也是遞迴可枚舉的；由上述定理， REJECT_{TM} 就會是可判定的。但這與 (a) 證得的不可判定性矛盾。

∴ $\overline{\text{REJECT}_{TM}}$ 不是遞迴可枚舉的。 ■

12 數學歸納法、DPLL 演算法、 $\text{SAT} \leq_p 3\text{SAT}$

[25 分]

題目說明

- a. [10 分] 給定遞迴關係 $T(0) = 0$ 、 $T(1) = 1$ 、 $T(n+1) = T(n) + 2T(n-1)$ ($n \geq 1$)。用對 n 的歸納法證明

$$T(n) = \frac{2^n - (-1)^n}{3} \quad (n \geq 0),$$

須清楚陳述歸納假設、基底情況與歸納步驟。

- b. [10 分] 用 DPLL 演算法判定下列子句集是否可滿足，並在每一步指明所用的規則：

$$(\neg P \vee R)_{(1)}, (\neg P \vee Q \vee T)_{(2)}, (Q \vee \neg R \vee \neg T)_{(3)}, (P \vee R \vee S)_{(4)}, (P \vee \neg R)_{(5)}, (P \vee R \vee \neg S)_{(6)}, (\neg P \vee \neg Q)_{(7)}$$

- c. [5 分] 描述一個把 SAT 實例轉為 3SAT 實例的多項式時間演算法，證明 SAT 可多項式歸約到 3SAT，並解釋演算法為何終止。

(a) 歸納法證明（10 分）

欲證命題： $P(n) : T(n) = \frac{2^n - (-1)^n}{3}$ ，對所有 $n \geq 0$ 成立。

由於遞迴式 $T(n+1) = T(n) + 2T(n-1)$ 同時用到前兩項，必須採用強歸納法（或稱二階歸納），且需要兩個基底情況。

基底情況：

$$n=0: \quad \frac{2^0 - (-1)^0}{3} = \frac{1-1}{3} = 0 = T(0). \checkmark \quad n=1: \quad \frac{2^1 - (-1)^1}{3} = \frac{2+1}{3} = 1 = T(1). \checkmark$$

歸納假設：設 $n \geq 1$ ，並假設 $P(n)$ 與 $P(n-1)$ 皆成立，即

$$T(n) = \frac{2^n - (-1)^n}{3}, \quad T(n-1) = \frac{2^{n-1} - (-1)^{n-1}}{3}.$$

歸納步驟：證明 $P(n+1)$ 。由遞迴式與歸納假設：

$$\begin{aligned} T(n+1) &= T(n) + 2T(n-1) \\ &= \frac{2^n - (-1)^n}{3} + 2 \cdot \frac{2^{n-1} - (-1)^{n-1}}{3} \\ &= \frac{2^n - (-1)^n + 2^n - 2(-1)^{n-1}}{3} = \frac{2^{n+1} - (-1)^n - 2(-1)^{n-1}}{3}. \end{aligned}$$

處理 (-1) 的幕次：因為 $(-1)^n = -(-1)^{n-1}$ ，

$$-(-1)^n - 2(-1)^{n-1} = (-1)^{n-1} - 2(-1)^{n-1} = -(-1)^{n-1} = (-1)^n = -(-1)^{n+1}.$$

代回得

$$T(n+1) = \frac{2^{n+1} - (-1)^{n+1}}{3},$$

即 $P(n+1)$ 成立。

結論：由兩個基底情況與歸納步驟，依強歸納法原理， $T(n) = \frac{2^n - (-1)^n}{3}$ 對所有 $n \geq 0$ 成立。

■

(b) DPLL 演算法 (10 分)

觀念補充

DPLL 的三個規則。(i) 單元傳播 (unit propagation)：若有單文字子句 (ℓ)，令 ℓ 為真——刪去所有含 ℓ 的子句、並從其餘子句刪去 $\neg \ell$ ；(ii) 純文字消去 (pure literal elimination)：若變數只以單一極性出現，令該極性為真並刪去相關子句；(iii) 分支 (case split / branching)：任選一變數，分別嘗試設真與設假。若某分支推導出空子句 $\square$ (矛盾) 則該分支失敗；所有分支皆失敗則不可滿足。

初始子句集沒有單元子句，也沒有純文字 (P, Q, R, S, T 皆同時以正、負極性出現)，因此先分支。

分支：選 P (case split)。

分支一：令 $P = \text{True}$ 。子句 (4)(5)(6) 因含 P 而被滿足刪除；(1)(2)(7) 中刪去 $\neg P$ ：

$$(R)_{(1')}, \quad (Q \vee T)_{(2')}, \quad (Q \vee \neg R \vee \neg T)_{(3)}, \quad (\neg Q)_{(7')}.$$

- 單元傳播 (R)：令 $R = \text{True}$ ，(3) 刪去 $\neg R$ 得 $(Q \vee \neg T)_{(3')}$ 。
- 單元傳播 ($\neg Q$)：令 $Q = \text{False}$ ，(2') 刪去 Q 得 (T) ；(3') 刪去 Q 得 $(\neg T)$ 。
- 單元傳播 (T)：令 $T = \text{True}$ ，子句 $(\neg T)$ 刪去 $\neg T$ 後變成空子句 $\square$ 。

得到矛盾，分支一失敗。

分支二：令 $P = \text{False}$ 。子句 (1)(2)(7) 因含 $\neg P$ 而被滿足刪除；(4)(5)(6) 中刪去 P ：

$$(R \vee S)_{(4')}, \quad (\neg R)_{(5')}, \quad (R \vee \neg S)_{(6')}.$$

- 單元傳播 $(\neg R)$ ：令 $R = \text{False}$ ，(4') 刪去 R 得 (S) ；(6') 刪去 R 得 $(\neg S)$ 。
- 單元傳播 (S) ：令 $S = \text{True}$ ，子句 $(\neg S)$ 變成空子句 $\square$ 。

得到矛盾，分支二失敗。

結論：變數 P 的兩個分支都推導出空子句，DPLL 回傳「不可滿足」。

$\therefore$ 該子句集 **Unsatisfiable** (不可滿足)。

(c) SAT $\leq_p$ 3SAT (5 分)

演算法 (子句切割法)。給定 SAT 實例 (CNF 公式) φ ，逐一處理每個子句 $c = (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k)$ ：

- 若 $k \leq 3$ ：以重複文字補足長度，例如 $(\ell_1) \mapsto (\ell_1 \vee \ell_1 \vee \ell_1)$ 、 $(\ell_1 \vee \ell_2) \mapsto (\ell_1 \vee \ell_2 \vee \ell_2)$ ，語意不變。
- 若 $k > 3$ ：引入 $k - 3$ 個全新變數 $z_1, \dots, z_{k-3}$ ，把 c 換成 $k - 2$ 個三元子句的串鏈：

$$(\ell_1 \vee \ell_2 \vee z_1) \wedge (\neg z_1 \vee \ell_3 \vee z_2) \wedge (\neg z_2 \vee \ell_4 \vee z_3) \wedge \cdots \wedge (\neg z_{k-3} \vee \ell_{k-1} \vee \ell_k).$$

正確性 (等可滿足性)。(⇒) 若原子句 c 被滿足，則某個 ℓ_i 為真；令 $z_1, \dots, z_{i-2}$ 為真、其餘 z_j 為假，則串鏈中每個三元子句都被滿足 (ℓ_i 之前的子句由 z_j 撐住，含 ℓ_i 的子句由 ℓ_i 撐住，其後的子句由 $\neg z_j$ 撐住)。(⇐) 反之，若所有 ℓ_i 皆為假，則串鏈化簡為 $(z_1) \wedge (\neg z_1 \vee z_2) \wedge \cdots \wedge (\neg z_{k-3})$ ，強迫 $z_1, z_2, \dots$ 逐一為真、最後卻要求 z_{k-3} 為假，矛盾——故串鏈可滿足必有某 ℓ_i 為真。因此新公式可滿足 $\iff \varphi$ 可滿足。

多項式時間與終止性。每個長度 $k > 3$ 的子句經一次處理後即被換成 $k - 2$ 個長度恰為 3 的子句，不會再被處理；演算法對每個子句只做一次轉換，處理的子句數量有限，故必定終止。輸出大小方面，長度 k 的子句產生 $O(k)$ 個新子句與 $O(k)$ 個新變數，總輸出大小與總時間皆為輸入大小的線性 (因此是多項式) 函數。故 SAT $\leq_p$ 3SAT。■

13 SAT 歸約到 CLIQUE 與 NP 成員資格

[25 分]

題目說明

CLIQUE 問題：輸入圖 $G = (V, E)$ 與整數 $k > 0$ ，若存在大小為 k 的集合 $C \subseteq V$ 使 C 中任兩頂點皆有邊相連，回傳 True。

- [4 分] 「SAT 可多項式歸約到 CLIQUE」是什麼意思？
- [9 分] 給定公式 $F = (\neg P \vee Q \vee \neg R) \wedge (P \vee Q) \wedge (\neg Q \vee R)$ ，建構圖 G_F ，使 F 可滿足 $\iff G_F$ 有大小 $k = 3$ 的 clique。
- [6 分] 解釋為何大小 $k = 3$ 的 clique 之存在保證 F 有滿足指派。
- [6 分] 描述一個多項式時間的非確定性演算法解 CLIQUE，即證明 CLIQUE $\in$ NP。

(a) 多項式歸約的定義 (4 分)

「SAT $\leq_p$ CLIQUE」的意思是：存在一個多項式時間可計算的轉換函數 f ，把每個 SAT 實例 (布林公式 φ) 映射成一個 CLIQUE 實例 $f(\varphi) = (G, k)$ ，並且滿足

$$\varphi \text{ 可滿足} \iff G \text{ 含有大小為 } k \text{ 的 clique.}$$

兩個要件缺一不可：(i) f 在輸入大小的多項式時間內可計算；(ii) 答案被雙向保持 (yes-instance 對應 yes-instance、no-instance 對應 no-instance)。其意義在於：若 CLIQUE 有多項式時間演算法，則透過 f ，SAT 也有；亦即 CLIQUE 至少與 SAT 一樣難。

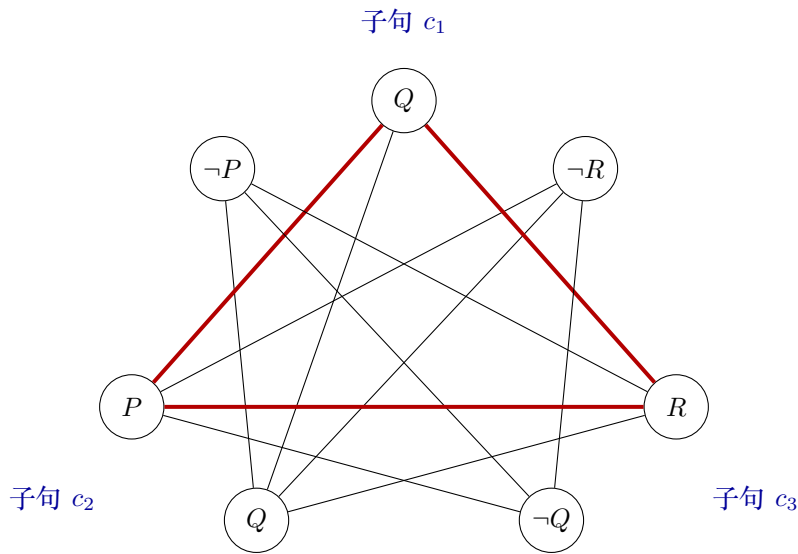
(b) 建構 G_F (9 分)

標準建構法：

1. 為每個子句中的每個文字出現建立一個頂點 (依子句分組)；
2. 兩頂點之間連邊 $\iff$ 它們 (i) 屬於不同子句，且 (ii) 不互為否定 (不是 X 與 $\neg X$ 的組合)；
3. 取 $k =$ 子句數。

F 有三個子句： $c_1 = (\neg P \vee Q \vee \neg R)$ 、 $c_2 = (P \vee Q)$ 、 $c_3 = (\neg Q \vee R)$ ，共 7 個文字出現， $k = 3$ 。

依規則逐對檢查 (同子句不連； $P/\neg P$ 、 $Q/\neg Q$ 、 $R/\neg R$ 不連) 得下圖：



邊的完整清單 (共 12 條)：

- c_1 - c_2 之間： $\neg P$ - Q_{c_2} 、 Q_{c_1} - P 、 Q_{c_1} - Q_{c_2} 、 $\neg R$ - P 、 $\neg R$ - Q_{c_2} (排除 $\neg P$ - P)；
- c_1 - c_3 之間： $\neg P$ - $\neg Q$ 、 $\neg P$ - R 、 Q_{c_1} - R 、 $\neg R$ - $\neg Q$ (排除 Q_{c_1} - $\neg Q$ 、 $\neg R$ - R)；
- c_2 - c_3 之間： P - $\neg Q$ 、 P - R 、 Q_{c_2} - R (排除 Q_{c_2} - $\neg Q$)。

圖中紅色粗邊即為一個大小 3 的 clique 例子： $\{Q_{c_1}, P_{c_2}, R_{c_3}\}$ (三條邊 Q - P 、 Q - R 、 P - R 都存在)。

(c) 為何 3-clique 保證 F 可滿足 (6 分)

設 $C = \{v_1, v_2, v_3\}$ 是 G_F 中大小為 3 的 clique。論證分三步：

(一) 三個頂點必來自三個不同子句。依建構，同一子句內的頂點之間沒有任何邊；clique 要求任兩點相鄰，故 C 不可能包含同子句的兩個頂點。因為恰有 $k = 3$ 個子句， C 必定在每個子句中恰取一個文字。

(二) 選出的文字彼此一致。依建構，互為否定的文字 (X 與 $\neg X$) 之間沒有邊；因此 C 中的三個文字兩兩不互相矛盾，可以同時為真。

(三) 由此得到滿足指派。定義指派：對 C 中的每個文字，若是正文字 X 則令 $X = \text{True}$ ，若是負文字 $\neg X$ 則令 $X = \text{False}$ ；未被提及的變數任意指定。由 (二) 此定義無衝突。由 (一) 每個子句都含有一個被設為真的文字，故每個子句都被滿足， F 為真。

以紅色 clique $\{Q_{c_1}, P_{c_2}, R_{c_3}\}$ 為例：令 $P = Q = R = \text{True}$ ，則 c_1 因 Q 為真、 c_2 因 P (及 Q) 為真、 c_3 因 R 為真而全部滿足。✓

(d) CLIQUE $\in$ NP (6 分)

非確定性多項式時間演算法：輸入 (G, k) ， $G = (V, E)$ ， $n = |V|$ 。

1. 猜測（非確定性步驟）：非確定性地選出一個大小為 k 的頂點子集 $C \subseteq V$ （例如對每個頂點非確定性地決定選或不選，再檢查 $|C| = k$ ）。
2. 驗證（確定性步驟）：對 C 中每一對頂點 $u \neq v$ ，檢查 $\{u, v\} \in E$ ；
3. 若所有 $\binom{k}{2}$ 對都相鄰則接受，否則此計算分支拒絕。

時間分析：猜測需要 $O(n)$ 個非確定性選擇；驗證需檢查 $\binom{k}{2} \leq \binom{n}{2} = O(n^2)$ 對頂點，每次相鄰性查詢在鄰接矩陣下為 $O(1)$ （鄰接串列下為 $O(n)$ ），總計至多 $O(n^3)$ ，是輸入大小的多項式。

正確性：若 G 有大小 k 的 clique，存在某個猜測分支恰好選中它並通過驗證（接受）；若沒有，所有分支都會在驗證中發現某對不相鄰的頂點而拒絕。故此非確定性演算法在多項式時間內判定 CLIQUE，即 CLIQUE $\in$ NP。■

（等價的證書觀點：certificate 就是頂點集 C 本身，長度多項式，驗證器在多項式時間內檢查 $|C| = k$ 且兩兩相鄰。）

選擇題答案總表 Q1: A Q2: D Q3: B, C Q4: B, E Q5: A
Q6: B Q7: A, E Q8: D Q9: B, C Q10: C