

King's College London

5CCS2FC2 Foundations of Computing II

2020 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 映射歸約（Mapping Reduction）的條件	[6 分] 3
3 Master 定理：判斷 $\Theta(n^2 \log n)$	[6 分] 4
4 2SAT 蘊涵圖的強連通分量個數	[3 分] 5
5 2SAT 可滿足性的充要條件	[3 分] 6
6 不可近似（Unapproximable）的定義	[4 分] 7
7 單體法（Simplex Method）的一次疊代	[7 分] 8
8 TSP：2-opt 疊代至局部最小	[6 分] 9
9 SCC 演算法：最後找到的強連通分量	[6 分] 10
10 Prim 演算法：邊 (E, G) 在第幾步加入	[5 分] 12
第二部分：長答題（第 11–13 題）	13
11 E_{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、分治法	[25 分] 14

13 HAMILTONIAN CYCLE : NP 成員資格與 SAT 歸約

[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

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

- A. 不存在對 X 而言 sound、complete 且 terminating 的演算法。
- B. 補問題 $\bar{X}$ 是可判定的（decidable）。
- C. 補問題 $\bar{X}$ 不是遞迴可枚舉的。
- D. 存在某個對 X 而言既 sound（可靠）又 complete（完備）的演算法。
- E. 以上皆非。

本題考三個核心性質的定義。設演算法 $\mathcal{A}$ 針對判定問題 X ：

- **Sound**（可靠）：若 $\mathcal{A}$ 回答「是」，則輸入真的屬於 X （不誤報）；
- **Complete**（完備）：對每個屬於 X 的輸入， $\mathcal{A}$ 都會回答「是」（不漏報）；
- **Terminating**（終止）： $\mathcal{A}$ 對所有輸入都在有限步內停止。

課程定義： X 可判定 $\iff$ 存在 sound + complete + terminating 的演算法； X 遞迴可枚舉 $\iff$ 存在 sound + complete 的演算法（不要求終止），即擁有「半判定程序」。

詳細解答

- **A.** 這是「 X 不可判定」的敘述，不是 r.e. 的定義。而且 r.e. 的問題也可能可判定（可判定 $\Rightarrow$ r.e.），此時 A 為假。錯誤。
- **B.** 若 $\bar{X}$ 可判定則 X 也可判定（把答案反過來即可），這比 r.e. 強得多；且存在 r.e. 但補問題不可判定的例子（如停機問題）。錯誤。
- **C.** 補問題是否 r.e. 與 X 本身是否 r.e. 是獨立的性質：可判定的 X 是 r.e. 且 $\bar{X}$ 也是 r.e.，此時 C 為假。錯誤。
- **D.** 「存在 sound 且 complete 的演算法」正是 r.e. 的定義。正確。
- **E.** 因 D 正確，排除。

正確答案

D

2 映射歸約（Mapping Reduction）的條件

[6 分]

題目說明

設 X, Y 是字母表 Σ 上的判定問題。函數 $f: \Sigma^* \rightarrow \Sigma^*$ 要成為從 X 到 Y 的映射歸約（mapping reduction，記作 $X \leq_m Y$ ），需要滿足哪些條件？（複選）

- A. $f(w) \in X$ 若且唯若 $f(w) \in Y$ 。
- B. $w \in X$ 若且唯若 $f(w) \in Y$ 。
- C. f 是多項式函數。
- D. f 是可計算（computable）函數。
- E. X 是不可判定的。

詳細解答

映射歸約的標準定義： $f: \Sigma^* \rightarrow \Sigma^*$ 是從 X 到 Y 的映射歸約，若

1. f 是可計算函數（存在圖靈機對每個輸入 w 都能停機並輸出 $f(w)$ ）；
2. 對所有 $w \in \Sigma^* : w \in X \iff f(w) \in Y$ （雙向保持答案：yes 對應 yes、no 對應 no）。

逐一檢視選項：

- **A.** 條件寫錯了：左邊應該是 $w \in X$ 而不是 $f(w) \in X$ 。「 $f(w) \in X \iff f(w) \in Y$ 」是關於輸出字串同時屬於兩個語言的怪異條件，與歸約無關。錯誤。
- **B.** 正是定義中的第 2 個條件。正確。
- **C.** 「多項式時間可計算」是多項式歸約 $\leq_p$ （用於 NP-completeness）的額外要求；一般的映射歸約（用於可計算性理論）只要求可計算，不要求多項式。錯誤。
- **D.** 正是定義中的第 1 個條件。若 f 不可計算，歸約就毫無用處（無法把「解 Y 的程序」轉成「解 X 的程序」）。正確。
- **E.** 映射歸約對任何一對語言都可以定義，與 X 是否可判定無關（例如可判定問題之間也常做歸約）。錯誤。

正確答案

B、D

3 Master 定理：判斷 $\Theta(n^2 \log n)$

[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 $\Theta(n^2 \log n)$ （皆假設 $T(1) = 1$ ）。（複選）

- A. $T(n) = 4T(n/2) + n \log_2 n$
- B. $T(n) = 4T(n/2) + n^2 \log_2 n$
- C. $T(n) = 9T(n/3) + n(n+1)$
- D. $T(n) = 2T(n/4) + 3n^2$
- E. 以上皆非。

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 $f(n) = O(n^{c^* - \epsilon})$ （某 $\epsilon > 0$ ），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**： $a = 4, b = 2, c^* = \log_2 4 = 2$ 。 $f(n) = n \log_2 n$ 。因為 $\log n$ 成長慢於任何正冪次， $n \log_2 n = O(n^{2-\epsilon})$ （取 $\epsilon = 0.5$ ），屬 **Case 1**：

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

選項 B： $a = 4, b = 2, c^* = 2$ 。 $f(n) = n^2 \log_2 n$ 。此函數恰好卡在縫隙：它不是 $O(n^{2-\varepsilon})$ 、不是 $\Theta(n^2)$ 、也不是 $\Omega(n^{2+\varepsilon})$ ($\log n$ 比任何 n^ε 都慢)，基本型 Master 定理三個 case 都不適用。用加強版 ($f(n) = \Theta(n^{c^*} \log^k n)$ 時 $T(n) = \Theta(n^{c^*} \log^{k+1} n)$ ，此處 $k = 1$) 可得

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

選項 C： $a = 9, b = 3, c^* = \log_3 9 = 2$ 。 $f(n) = n(n+1) = n^2 + n = \Theta(n^2)$ ，恰等於臨界冪次，屬 Case 2：

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

選項 D： $a = 2, b = 4, c^* = \log_4 2 = \frac{1}{2}$ 。 $f(n) = 3n^2 = \Omega(n^{1/2+\varepsilon})$ ，檢查正則條件：

$$a f(n/b) = 2 \cdot 3 \left(\frac{n}{4}\right)^2 = \frac{3}{8} n^2 = \frac{1}{8} \cdot 3n^2 \leq c f(n), \quad c = \frac{1}{8} < 1. \quad \checkmark$$

屬 Case 3： $T(n) = \Theta(n^2) \neq \Theta(n^2 \log n)$ 。不符。

正確答案

C

4 2SAT 蘊涵圖的強連通分量個數

[3 分]

題目說明

考慮以下 2SAT 實例：

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

其蘊涵圖 (implication graph) 中共有幾個強連通分量 (SCC)？(單選)

- A. 1 B. 2 C. 3 D. 4 E. 5 或以上

觀念補充

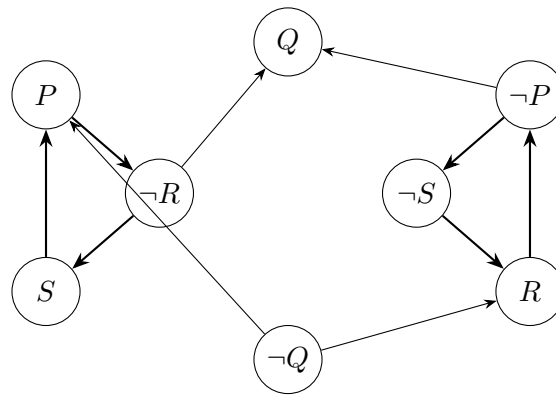
蘊涵圖的建構。每個子句 $(a \vee b)$ 邏輯上等價於兩條蘊涵： $\neg a \rightarrow b$ 與 $\neg b \rightarrow a$ 。蘊涵圖以全部 $2n$ 個文字（每個變數及其否定）為節點，為每個子句加入這兩條有向邊。

詳細解答

步驟一：由每個子句寫出兩條蘊涵邊。

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

步驟二：畫出蘊涵圖。(8 個節點、10 條邊)



步驟三：找出強連通分量。

- 迴圈 $P \rightarrow \neg R \rightarrow S \rightarrow P$: $\{P, \neg R, S\}$ 互相可達，構成一個 SCC。
- 鏡像迴圈 $\neg P \rightarrow \neg S \rightarrow R \rightarrow \neg P$: $\{\neg P, \neg S, R\}$ 構成一個 SCC (2SAT 蘊涵圖的對偶對稱性)。
- Q 沒有任何出邊 (Q 只出現在子句的正文字位置，其蘊涵都指向 Q)，到不了任何點，故 $\{Q\}$ 自成單點 SCC。
- $\neg Q$ 沒有任何入邊 (沒有蘊涵以 $\neg Q$ 為終點)，沒有點能回到它，故 $\{\neg Q\}$ 自成單點 SCC。

共 $\{P, \neg R, S\}$ 、 $\{\neg P, \neg S, R\}$ 、 $\{Q\}$ 、 $\{\neg Q\}$ 四個 SCC。

(附帶一提：沒有任何變數與其否定同處一個 SCC，故此公式可滿足；例如 $P = \text{True}, R = \text{False}, S = \text{True}, Q = \text{True}$ 。)

正確答案

D (4 個)

5 2SAT 可滿足性的充要條件

[3 分]

題目說明

下列關於蘊涵圖的哪一個條件，是 2SAT 實例可滿足的充分且必要條件？(單選)

- 任何文字與其否定之間都沒有路徑。
- 每個強連通分量只含正文字、或只含負文字。
- 不存在同時包含某文字及其否定的迴圈 (cycle)。
- 每個強連通分量都非空。
- 以上皆非。

詳細解答

標準判準 (Aspvall–Plass–Tarjan 定理)：

2SAT 可滿足 $\iff$ 沒有任何變數 X 使 X 與 $\neg X$ 落在同一個 SCC 中。

而「 X 與 $\neg X$ 在同一個 SCC」 $\iff$ 「存在一條同時經過 X 與 $\neg X$ 的迴圈」(互相可達即可沿 $X \rightsquigarrow \neg X \rightsquigarrow X$ 繞一圈)。因此選項 C 與標準判準等價。

逐一檢視：

- **A.** 太強了，只是充分條件而非必要。單向路徑 $X \rightsquigarrow \neg X$ 是允許的——它只是強迫 X 必須為假。例如公式 $(\neg X \vee \neg X)$ 給出邊 $X \rightarrow \neg X$ ，公式可滿足 (X 取假)，但 A 的條件不成立。錯誤。

- **B.** 既不必要也不充分。第 4 題的 SCC $\{P, \neg R, S\}$ 混合了正負文字，而該公式可滿足——不必要。錯誤。
- **C.** 「沒有迴圈同時含 X 與 $\neg X$ 」 $\iff$ 「 X 與 $\neg X$ 不同 SCC (對所有 X)」 $\iff$ 可滿足。充要，正確。
- **D.** 空集合本來就不會是強連通分量 (SCC 是非空的極大集合)，此條件恆真，對任何公式都成立，不能區分可滿足與否。錯誤。

正確答案

C

6 不可近似 (Unapproximable) 的定義

[4 分]

題目說明

「最佳化問題 X 是 **unapproximable** (不可近似，假設 $P \neq NP$)」是什麼意思？(單選)

- A. 不存在多項式時間演算法能在至多 r 次疊代後回傳最佳解。
- B. 存在近似比 $r < 1$ 的多項式時間近似演算法。
- C. 對任何固定的 $r \geq 1$ ，都不存在近似比為 r 的多項式時間近似演算法。
- D. X 對應的判定問題是不可判定的。
- E. 以上皆非。

詳細解答

回顧定義： X 是 **r -approximable**，若存在多項式時間演算法回傳的解保證不劣於最佳解的 r 倍。「unapproximable」是它的全稱否定：

對每一個固定常數 $r \geq 1$ ， X 都不是 r -approximable。

也就是說，除非 $P = NP$ ，任何多項式時間演算法都無法給出任何常數倍品質保證。經典例子：一般（不滿足三角不等式的）TSP 就是 unapproximable——若它有任何常數比的近似演算法，就能用來在多項式時間內解 Hamiltonian Cycle，導致 $P = NP$ 。

逐一檢視：

- **A.** 把 r 誤解為「疊代次數」； r 是解品質的倍率，且此敘述與近似性無關。錯誤。
- **B.** 近似比依定義 ≥ 1 ($r = 1$ 表示精確最佳)； $r < 1$ 表示「比最佳解更好」，不可能。錯誤。
- **C.** 正是 unapproximable 的定義。正確。
- **D.** 混淆了「計算困難 (NP-hard)」與「不可判定」。NP 最佳化問題的判定版本都是可判定的（窮舉即可），只是慢。錯誤。

正確答案

C

7 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：

$$T_1 =$$

x	y	s_1	s_2	s_3	C	
7	3	1	0	0	0	-8
1	-8	0	1	0	0	5
1	-7	0	0	1	0	9
-3	1	0	0	0	1	0

找出樞軸 (pivot)，對 T_1 執行一次單體法疊代得到 T_2 。依題目指定的位置讀出： Z (第 1 列 s_2 行)、 X (第 2 列 x 行)、 W (第 3 列最右行)、 Y (第 4 列 y 行)。(單選)

- A. $X = -6/7, Y = 0, Z = -3, W = -9/7$
 B. $X = 0, Y = 2/7, Z = 0, W = 59/7$
 C. $X = 1, Y = -23, Z = -7, W = 4$
 D. $X = 0, Y = -7, Z = 1, W = 59$
 E. 以上皆非。

詳細解答

步驟一：選樞軸行 (entering variable)。目標函數列為 $(-3, 1, 0, 0, 0)$ ，唯一的負係數是 x 行的 -3 ，故選 x 行。

步驟二：比值測試 (ratio test)。對樞軸行中為正的元素計算「右端值 $\div$ 該元素」，取最小的非負比值：

列	x 行元素	比值	判斷
1	7	$-8/7 \approx -1.14$	比值為負，不考慮
2	1	$5/1 = 5$	候選 (最小)
3	1	$9/1 = 9$	候選

最小非負比值在第 2 列，故樞軸元素為第 2 列 x 行的 1。

步驟三：樞軸列正規化。樞軸元素已是 1，不需縮放：

$$R'_2 = (1, -8, 0, 1, 0, 0 \mid 5).$$

步驟四：消去其他列的 x 行。

$$R'_1 = R_1 - 7R'_2 = (7 - 7, 3 + 56, 1, -7, 0, 0 \mid -8 - 35) = (0, 59, 1, -7, 0, 0 \mid -43),$$

$$R'_3 = R_3 - R'_2 = (0, -7 + 8, 0, -1, 1, 0 \mid 9 - 5) = (0, 1, 0, -1, 1, 0 \mid 4),$$

$$R'_4 = R_4 + 3R'_2 = (-3 + 3, 1 - 24, 0, 3, 0, 1 \mid 0 + 15) = (0, -23, 0, 3, 0, 1 \mid 15).$$

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	59	1	-7	0	0	-43
1	-8	0	1	0	0	5
0	1	0	-1	1	0	4
0	-23	0	3	0	1	15

Z (第 1 列 s_2 行) = -7； X (第 2 列 x 行) = 1； W (第 3 列右端) = 4； Y (第 4 列 y 行) = -23。

正確答案

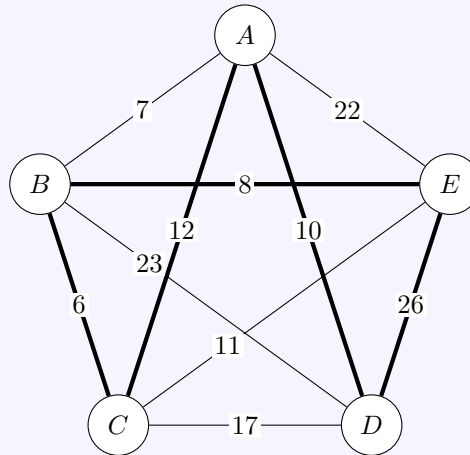
C ($X = 1, Y = -23, Z = -7, W = 4$)

8 TSP : 2-opt 疊代至局部最小

[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。反覆套用 2-opt 演算法直到抵達局部最小。需要套用幾次 2-opt 交換規則？(單選)

- A. 1 B. 2 C. 3 D. 4 E. 5 次或以上

觀念補充

2-opt 交換。從迴路中移除兩條不相鄰的邊，迴路斷成兩段，再以唯一另一種方式重新接合 (等價於反轉其中一段)。五個城市的迴路恰有 5 對不相鄰的邊，故每條路線恰有 5 個 2-opt 鄰居。演算法反覆執行能使路線變短的交換，直到沒有任何交換能再改善 (局部最小)。

詳細解答

權重表：

	A	B	C	D	E
A	—	7	12	10	22
B	7	—	6	23	8
C	12	6	—	17	11
D	10	23	17	—	26
E	22	8	11	26	—

目前路線 $A \rightarrow C \rightarrow B \rightarrow E \rightarrow D \rightarrow A$: $12 + 6 + 8 + 26 + 10 = 62$ 。✓

第一輪：檢查目前路線 (62) 的全部 5 個 **2-opt** 鄰居。每個交換「移除兩條邊、加入兩條邊」的淨變化為 $\Delta = (\text{新邊和}) - (\text{舊邊和})$ ：

移除	加入	新路線	長度	Δ
$(A, C), (B, E)$	$(A, B), (C, E)$	$A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A$	$7 + 6 + 11 + 26 + 10 = 60$	-2
$(A, C), (E, D)$	$(C, D), (A, E)$	$A \rightarrow E \rightarrow B \rightarrow C \rightarrow D \rightarrow A$	63	+1
$(C, B), (E, D)$	$(C, E), (B, D)$	$A \rightarrow C \rightarrow E \rightarrow B \rightarrow D \rightarrow A$	64	+2
$(C, B), (D, A)$	$(C, D), (B, A)$	$A \rightarrow B \rightarrow E \rightarrow D \rightarrow C \rightarrow A$	70	+8
$(B, E), (D, A)$	$(B, D), (E, A)$	$A \rightarrow C \rightarrow B \rightarrow D \rightarrow E \rightarrow A$	89	+27

唯一能改善的交換是第一列：第 1 次套用，得到新路線 $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A$ ，長度 60。

第二輪：檢查路線 (60) 的 5 個鄰居。

移除	加入	新路線	長度	Δ
$(A, B), (C, E)$	$(A, C), (B, E)$	回到原路線	62	+2
$(A, B), (E, D)$	$(B, D), (A, E)$	$A \rightarrow D \rightarrow B \rightarrow C \rightarrow E \rightarrow A$	72	+12
$(B, C), (E, D)$	$(C, D), (B, E)$	$A \rightarrow B \rightarrow E \rightarrow C \rightarrow D \rightarrow A$	$7+8+11+17+10 = 53$	-7
$(B, C), (D, A)$	$(A, C), (B, D)$	$A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow A$	79	+19
$(C, E), (D, A)$	$(C, D), (A, E)$	$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow A$	78	+18

唯一能改善的交換是第三列：第 2 次套用，得到 $A \rightarrow B \rightarrow E \rightarrow C \rightarrow D \rightarrow A$ ，長度 53。

第三輪：檢查路線 (53) 的 5 個鄰居。

移除 $(A, B), (E, C) \Rightarrow 63$; 移除 $(A, B), (C, D) \Rightarrow 64$;
 移除 $(B, E), (C, D) \Rightarrow 60$; 移除 $(B, E), (D, A) \Rightarrow 78$;
 移除 $(E, C), (D, A) \Rightarrow 70$.

所有鄰居都 ≥ 53 ，沒有任何改善交換，已抵達局部最小（事實上 53 也是全域最佳解）。

結論：62 $\xrightarrow{\text{第 1 次}}$ 60 $\xrightarrow{\text{第 2 次}}$ 53（停止）。共需 2 次 2-opt 交換。由於每一輪的改善交換都是唯一的，這個次數與交換的選取策略無關。

正確答案

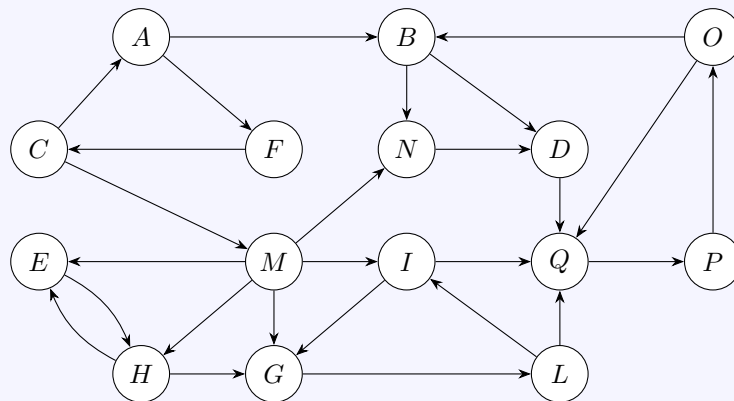
B (2 次)

9 SCC 演算法：最後找到的強連通分量

[6 分]

題目說明

考慮以下相依圖 (dependency graph)：



套用強連通分量 (SCC) 演算法找出所有的 SCC。演算法最後找到的 SCC 是哪一個？(單選)

- A. $\{E, H\}$ B. $\{M\}$ C. $\{A, C, F\}$ D. $\{B, D, N, O, P, Q\}$ E. 以上皆非。

觀念補充

課程中的 SCC 演算法 (Kosaraju 式)。(1) 對 G 做 DFS 產生「拓撲式排序」清單 L (依完成時間由晚到早)；(2) 依 L 的順序，每次取出清單最前面尚未歸類的節點 u ，在轉置圖 G^T (所有邊反向) 上從 u 做 DFS，走到的所有未歸類節點構成一個 SCC；(3) 重複直到清單為空。可以證明：各 SCC 被找到的順序，正是元件圖 (condensation) 的一個拓撲順序——由「源頭」SCC 往「匯點」SCC 依序輸出。

詳細解答

步驟一：先確定 SCC 的劃分（沿有向迴圈找互相可達的集合）：

- $A \rightarrow F \rightarrow C \rightarrow A : S_1 = \{A, C, F\}$;
- $C \rightarrow M$, 而 M 沒有任何路徑回到 $C : S_2 = \{M\}$;
- $E \rightarrow H$ 與 $H \rightarrow E : S_3 = \{E, H\}$;
- $G \rightarrow L \rightarrow I \rightarrow G : S_4 = \{G, I, L\}$;
- $B \rightarrow N \rightarrow D \rightarrow Q \rightarrow P \rightarrow O \rightarrow B : S_5 = \{B, D, N, O, P, Q\}$ 。

共 5 個 SCC ($3 + 1 + 2 + 3 + 6 = 15$ 個節點 $\checkmark$) 。

步驟二：模擬演算法（節點選取採字母順序）。第一趟在 G 上做 DFS（從 A 出發），完成順序為

$O, P, Q, D, N, B, I, L, G, H, E, M, C, F, A,$

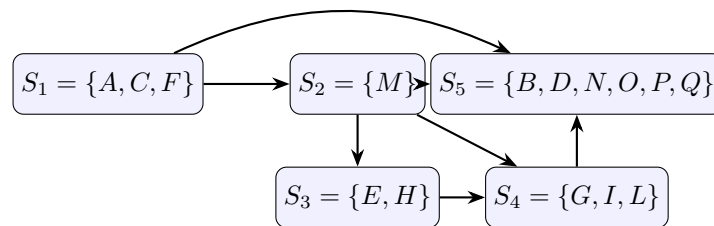
故清單（完成得越晚排越前面）為

$L = \langle A, F, C, M, E, H, G, L, I, B, N, D, Q, P, O \rangle.$

步驟三：依序在轉置圖 G^T 上做 DFS 。

1. 取 $A : G^T$ 中 $A \leftarrow C \leftarrow F \leftarrow A$, 找到 $S_1 = \{A, C, F\}$;
2. 取 $M : M$ 在 G^T 中唯一的鄰居 C 已歸類, 找到 $S_2 = \{M\}$;
3. 取 $E : E \leftrightarrow H$, 找到 $S_3 = \{E, H\}$;
4. 取 $G : G \leftarrow I \leftarrow L \leftarrow G$, 找到 $S_4 = \{G, I, L\}$;
5. 取 $B : 沿 G^T 走遍 B, O, P, Q, D, N$, 找到 $S_5 = \{B, D, N, O, P, Q\}$ 。（最後）

為什麼答案不依賴節點選取順序？元件圖如下：



演算法輸出 SCC 的順序必為元件圖的拓撲順序，而 S_5 是元件圖中唯一的匯點（沒有出邊： S_5 內的 B, D, N, O, P, Q 的所有出邊都留在 S_5 內部），因此不論 DFS 的起點與鄰居順序如何選， $S_5 = \{B, D, N, O, P, Q\}$ 必定是最後被找到的 SCC 。

正確答案

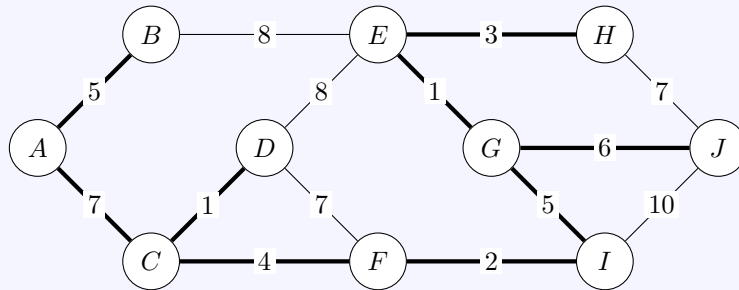
D ($\{B, D, N, O, P, Q\}$)

10 Prim 演算法：邊 (E, G) 在第幾步加入

[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）：



從頂點 A 開始，用 **Prim** 演算法建構 G 的最小生成樹。邊 (E, G) 在第幾步被加入生成樹？（單選）

- A. 1 B. 3 C. 5 D. 7 E. 以上皆非。

觀念補充

Prim 演算法。從起點出發維護一棵樹 T ；每一步在所有「一端在 T 內、另一端在 T 外」的跨越邊中，選權重最小者加入 T ，直到 T 包含全部頂點（ $n - 1 = 9$ 步）。

詳細解答

全部 14 條邊： $AB=5$, $AC=7$, $BE=8$, $CD=1$, $CF=4$, $DE=8$, $DF=7$, $EG=1$, $EH=3$, $FI=2$, $GI=5$, $GJ=6$,
從 A 開始逐步執行（每步列出所有跨越邊，粗體為選中的最小者）：

步驟	跨越邊（權重）	加入的邊	樹中頂點
1	AB(5) , AC(7)	$A-B$	$\{A, B\}$
2	AC(7) , BE(8)	$A-C$	$\{A, B, C\}$
3	BE(8), CD(1) , CF(4)	$C-D$	$\{A, B, C, D\}$
4	BE(8), CF(4) , DE(8), DF(7)	$C-F$	$\{A, B, C, D, F\}$
5	BE(8), DE(8), FI(2)	$F-I$	$\{A, B, C, D, F, I\}$
6	BE(8), DE(8), GI(5) , IJ(10)	$G-I$	$\{A, B, C, D, F, G, I\}$
7	BE(8), DE(8), EG(1) , GJ(6), IJ(10)	E-G	$\{A, \dots, G, I\} \cup \{E\}$
8	EH(3) , GJ(6), IJ(10)	$E-H$	$+H$
9	GJ(6) , HJ(7), IJ(10)	$G-J$	全部 10 個頂點 ✓

說明幾個關鍵步驟：

- 第 3 步起 $CD=1$ 是全場最小的跨越邊，優先於 $BE(8)$ ；
- 第 5、6 步沿著下排 $F \rightarrow I \rightarrow G$ 便宜地擴張（2 與 5 都小於 8）；
- 第 7 步： G 進入樹後， $EG=1$ 成為新的跨越邊且是目前最小者，此時才把 E 拉進樹——邊 (E, G) 於第 7 步加入；
- 每一步的最小跨越邊都是唯一的，因此答案不受平手規則影響。

（檢驗：MST 邊為 $AB, AC, CD, CF, FI, GI, EG, EH, GJ$ ，共 9 條，總權重 $5+7+1+4+2+5+1+3+6 = 34$ 。）

正確答案

D（第 7 步）

第二部分：長答題（第 11–13 題，任選兩題，每題 25 分）

11 E_{TM} 的不可判定性與遞迴可枚舉性

[25 分]

題目說明

考慮「語言非空」的圖靈機編碼語言

$$E_{TM} = \{\langle M \rangle : M \text{ 編碼一台圖靈機且 } L(M) \neq \emptyset\}$$

（即 M 至少接受一個字串）。

- [13 分] 建構一個從 A_{TM} 到 E_{TM} 的映射歸約，證明 E_{TM} 不可判定；其中 $A_{TM} = \{\langle M, w \rangle : M \text{ 編碼一台接受字串 } w \text{ 的圖靈機}\}$ ，可假設 A_{TM} 已知不可判定。
- [7 分] 建構一個 sound 且 complete 的演算法辨識所有 $\langle M \rangle \in E_{TM}$ ，證明 E_{TM} 是遞迴可枚舉的。
- [5 分] 據此（或以其他方式）證明補語言 $\overline{E_{TM}}$ 不是遞迴可枚舉的。

(a) 映射歸約 $A_{TM} \leq_m E_{TM}$ (13 分)

歸約函數的建構。定義 f ：給定輸入 $\langle M, w \rangle$ ，輸出一台新圖靈機的編碼 $\langle M_w \rangle$ ，其中 M_w 的行為如下：

機器 M_w ：輸入任意字串 x 。

- 忽略 x ；
- 在內部模擬 M 於固定字串 w 上的執行（ M 與 w 都被寫死在 M_w 的程式裡）；
- 若 M 接受 w ： M_w 接受 x ；若 M 拒絕 w ： M_w 拒絕 x （若 M 不停機， M_w 也不停機）。

f 是可計算的。 f 只做語法上的組裝：把 M 的程式碼與常數字串 w 嵌進一個固定模板（「忽略輸入、模擬 M 於 w 」）。這不需要執行 M ，只是文字替換，顯然有圖靈機能對每個輸入 $\langle M, w \rangle$ 在有限時間內輸出 $\langle M_w \rangle$ 。✓

答案的雙向保持。分析 M_w 的語言：

$$L(M_w) = \begin{cases} \Sigma^*, & \text{若 } M \text{ 接受 } w \text{ (每個 } x \text{ 都被接受),} \\ \emptyset, & \text{若 } M \text{ 不接受 } w \text{ (沒有任何 } x \text{ 被接受).} \end{cases}$$

因此

$$\langle M, w \rangle \in A_{TM} \iff L(M_w) = \Sigma^* \neq \emptyset \iff \langle M_w \rangle \in E_{TM}. \checkmark$$

兩個條件都滿足， f 是從 A_{TM} 到 E_{TM} 的映射歸約，即 $A_{TM} \leq_m E_{TM}$ 。

導出不可判定性。反設 E_{TM} 可判定，設 R 是其總是停機的判定器。則以下程序判定 A_{TM} ：輸入 $\langle M, w \rangle$ ，計算 $f(\langle M, w \rangle) = \langle M_w \rangle$ （有限時間），執行 $R(\langle M_w \rangle)$ 並回傳其答案。此程序 sound、complete 且終止，即 A_{TM} 可判定——與已知的 A_{TM} 不可判定矛盾。

∴ E_{TM} 不可判定。 ■

(b) 遞迴可枚舉性：交錯模擬 (7 分)

難點：要判斷 $L(M) \neq \emptyset$ 得「找到一個被接受的字串」，但我們既不知道是哪個字串，也不知道要模擬多少步；若天真地逐字串完整模擬，可能卡死在某個不停機的字串上。解法是交錯模擬 (dovetailing)：

演算法 $\mathcal{S}$ ：輸入 $\langle M \rangle$ 。設 $w_1, w_2, w_3, \dots$ 是 Σ^* 的（可計算的）標準枚舉。

1. 對 $k = 1, 2, 3, \dots$ ：
2. 對前 k 個字串 $w_1, \dots, w_k$ ，各模擬 M 至多 k 步；
3. 若其中任何一次模擬中 M 接受了某個 w_i ： $\mathcal{S}$ 回答「接受」並停止；
4. 否則 $k += 1$ 繼續（可能永遠進行下去）。

Soundness： $\mathcal{S}$ 只在模擬中親眼看到 M 接受某字串 w_i 時才回答「接受」，此時 $w_i \in L(M)$ ，故 $L(M) \neq \emptyset$ ， $\langle M \rangle \in E_{TM}$ 。不誤報。✓

Completeness：若 $\langle M \rangle \in E_{TM}$ ，則存在某字串被接受；設它在枚舉中是第 i 個（ w_i ），且 M 在 n_i 步內接受它。取 $k^* = \max(i, n_i)$ ：在第 k^* 輪， $\mathcal{S}$ 會把 w_i 模擬至少 n_i 步，必然看到接受並回答「接受」。不漏報。✓

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

（注意 $\mathcal{S}$ 不終止：當 $L(M) = \emptyset$ 時它永遠繞圈——這與 (a) 的不可判定性一致。）

(c) $\overline{E_{TM}}$ 不是遞迴可枚舉的（5 分）

觀念補充

關鍵定理。若 X 與 $\overline{X}$ 都是遞迴可枚舉的，則 X 可判定。

證明概要：設 S_1, S_2 分別是 $X, \overline{X}$ 的半判定演算法。給定輸入 x ，交錯地並行模擬 $S_1(x)$ 與 $S_2(x)$ （輪流各走一步）。 x 必屬於 X 或 $\overline{X}$ 之一，由完備性其中一個模擬必在有限步內接受；依先接受者回答「 $x \in X$ 」或「 $x \notin X$ 」。此程序 sound、complete 且終止。

反設 $\overline{E_{TM}}$ 是遞迴可枚舉的。由 (b)， E_{TM} 也是遞迴可枚舉的；由上述定理， E_{TM} 便是可判定的——與 (a) 的結論矛盾。

∴ $\overline{E_{TM}}$ 不是遞迴可枚舉的。 ■

12 數學歸納法、DPLL 演算法、分治法

[25 分]

題目說明

- a. [10 分] 給定遞迴關係 $T(1) = 1, T(n) = T(\lceil n/3 \rceil) + 2$ ($n > 1$)。用對 n 的歸納法證明 $T(n) \geq \log_2(n)$ 對所有 $n \geq 1$ 成立。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：

$$\begin{array}{llll} (Q \vee \neg S \vee \neg T)_{(1)} & (\neg Q \vee \neg T)_{(2)} & (\neg P \vee R \vee \neg S \vee \neg T)_{(3)} & (P \vee Q \vee S)_{(4)} \\ (P \vee \neg S \vee T)_{(5)} & (\neg P \vee Q)_{(6)} & (\neg Q \vee T)_{(7)} & \end{array}$$

- c. [5 分] 描述一個你自選問題的分治（Divide-and-Conquer）演算法，以清晰度計分。

(a) 歸納法證明 $T(n) \geq \log_2 n$ (10 分)

欲證命題： $P(n) : T(n) \geq \log_2(n)$ ，對所有 $n \geq 1$ 。

由於遞迴式把 n 送到 $\lceil n/3 \rceil$ （不是 $n-1$ ），採用強歸納法。

基底情況 ($n = 1$)：

$$T(1) = 1 \geq 0 = \log_2(1). \checkmark$$

歸納假設：設 $n > 1$ ，並假設對所有 $1 \leq m < n$ 皆有 $T(m) \geq \log_2(m)$ 。

歸納步驟：先確認遞迴引用的參數落在假設範圍內：對 $n > 1$ ，

$$1 \leq \left\lceil \frac{n}{3} \right\rceil \leq \frac{n+2}{3} < n \quad (\text{末一步因 } n+2 < 3n \iff n > 1),$$

故可對 $m = \lceil n/3 \rceil$ 使用歸納假設。於是

$$\begin{aligned} T(n) &= T\left(\left\lceil \frac{n}{3} \right\rceil\right) + 2 \geq \log_2\left(\left\lceil \frac{n}{3} \right\rceil\right) + 2 && (\text{歸納假設}) \\ &\geq \log_2\left(\frac{n}{3}\right) + 2 && (\lceil n/3 \rceil \geq n/3, \log_2 \text{ 遞增}) \\ &= \log_2(n) - \log_2(3) + 2 \\ &\geq \log_2(n) && (\log_2 3 < \log_2 4 = 2). \end{aligned}$$

即 $P(n)$ 成立。

結論：由基底情況與歸納步驟，依強歸納法原理， $T(n) \geq \log_2(n)$ 對所有 $n \geq 1$ 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)：若某變數在全部子句中只以單一極性出現，令該極性為真並刪去含它的子句；(iii) 分支 (case split)：任選一變數分別嘗試設真、設假；分支中推導出空子句 $\square$ 即失敗，所有分支皆失敗則不可滿足。

初始檢查。沒有單元子句。檢查純文字： P (正：4, 5；負：3, 6) 混合； Q (正：1, 4, 6；負：2, 7) 混合； S 、 T 皆混合；但 R 只以正文字出現 (僅在子句 (3))。

步驟 1 (純文字消去 R)：令 $R = \text{True}$ ，刪去子句 (3)。剩餘：

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

仍無單元子句、無純文字。

步驟 2 (分支：選 Q)。

分支一： $Q = \text{True}$ 。含 Q 的 (1)(4)(6) 被滿足刪除；(2)(7) 刪去 $\neg Q$ ：

$$(\neg T)_{(2')}, \quad (P \vee \neg S \vee T)_{(5)}, \quad (T)_{(7')}.$$

• 單元傳播 ($\neg T$)：令 $T = \text{False}$ ；子句 $(T)_{(7')}$ 刪去 T 後成為空子句 $\square$ 。

矛盾，分支一失敗。(本質原因： $(\neg Q \vee \neg T)$ 與 $(\neg Q \vee T)$ 在 Q 為真時同時要求 $\neg T$ 與 T)

分支二： $Q = \text{False}$ 。含 $\neg Q$ 的 (2)(7) 被滿足刪除；(1)(4)(6) 刪去 Q ：

$$(\neg S \vee \neg T)_{(1')}, \quad (P \vee S)_{(4')}, \quad (P \vee \neg S \vee T)_{(5)}, \quad (\neg P)_{(6')}.$$

• 單元傳播 ($\neg P$)：令 $P = \text{False}$ ；(4') 刪去 P 得 (S) ；(5) 刪去 P 得 $(\neg S \vee T)$ 。

• 單元傳播 (S)：令 $S = \text{True}$ ；(1') 刪去 $\neg S$ 得 $(\neg T)$ ； $(\neg S \vee T)$ 刪去 $\neg S$ 得 (T) 。

• 單元傳播 ($\neg T$)：令 $T = \text{False}$ ；子句 (T) 成為空子句 $\square$ 。

矛盾，分支二失敗。

結論：變數 Q 的兩個分支都推導出空子句，DPLL 回傳

Unsatisfiable (不可滿足)。

(快速驗證：(2) 與 (7) 強迫 Q 為假；(6) 再強迫 P 為假；(4) 強迫 S 為真；(5) 強迫 T 為真；但此時 (1) = $Q \vee \neg S \vee \neg T$ 三個文字全假——的確無解。)

(c) 分治演算法範例：合併排序 (Merge Sort) (5 分)

問題：給定長度 n 的數列 $a_1, \dots, a_n$ ，輸出由小到大排序後的數列。

演算法 (分治三部曲)：

1. **Divide** (分割)：若 $n \leq 1$ 直接回傳 (已排序)。否則把數列切成左半 $a_1 \dots a_{\lfloor n/2 \rfloor}$ 與右半 $a_{\lfloor n/2 \rfloor + 1} \dots a_n$ ；
2. **Conquer** (征服)：對左、右兩半遞迴地呼叫合併排序，得到兩個已排序的子數列；
3. **Combine** (合併)：用兩根指標同時從兩個已排序子數列的開頭掃描，每次把較小的元素搬進輸出，直到兩邊都取完——線性時間 $O(n)$ 完成合併。

正確性：對長度做歸納——長度 ≤ 1 顯然正確；假設遞迴呼叫正確排序兩半，合併步驟按大小交錯取出即得整體排序。

時間複雜度：遞迴關係為

$$T(n) = 2T(n/2) + O(n),$$

$a = 2, b = 2, c^* = \log_2 2 = 1, f(n) = O(n) = \Theta(n^{c^*})$ ，Master 定理 **Case 2** 給出 $T(n) = \Theta(n \log n)$ ——優於插入排序等 $\Theta(n^2)$ 演算法。

13 HAMILTONIAN CYCLE：NP 成員資格與 SAT 歸約 [25 分]

題目說明

HAMILTONIAN CYCLE 問題：輸入圖 $G = (V, E)$ ，若存在一條恰好經過每個頂點一次的迴路，回傳 True。

- a. [6 分] 描述一個多項式時間的非確定性演算法，證明 HAMILTONIAN CYCLE $\in$ NP。
- b. [9 分] 給定命題公式 F ，描述建構圖 G_F 的程序，使得 F 可滿足 $\iff G_F$ 有 Hamiltonian cycle。
- c. [5 分] 解釋為何 F 可滿足保證 G_F 有 Hamiltonian cycle。
- d. [5 分] 解釋為何 G_F 有 Hamiltonian cycle 保證 F 可滿足。

(a) HAMILTONIAN CYCLE $\in$ NP (6 分)

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

1. 猜測 (非確定性步驟)：非確定性地選出頂點的一個排列 $\langle v_1, v_2, \dots, v_n \rangle$ ；
2. 驗證 (確定性步驟)：
 - 檢查 $v_1, \dots, v_n$ 兩兩相異且涵蓋全部 V (即為一個排列)；
 - 對每個 $i = 1, \dots, n-1$ 檢查 $\{v_i, v_{i+1}\} \in E$ ，並檢查 $\{v_n, v_1\} \in E$ (收尾成迴路)；
3. 全部通過則接受，否則此分支拒絕。

時間分析：猜測需 $O(n \log n)$ 個非確定性位元；驗證做 n 次相鄰性檢查與一次排列檢查，鄰接矩陣下共 $O(n^2)$ 。多項式。✓

正確性：若 G 有 Hamiltonian cycle，存在某個猜測分支恰好列出該迴路的頂點順序並通過驗證；若沒有，所有分支都會在驗證中失敗。故此演算法在非確定性多項式時間內判定 HAMILTONIAN CYCLE，即它屬於 NP。■

(b) 建構 G_F (9 分)

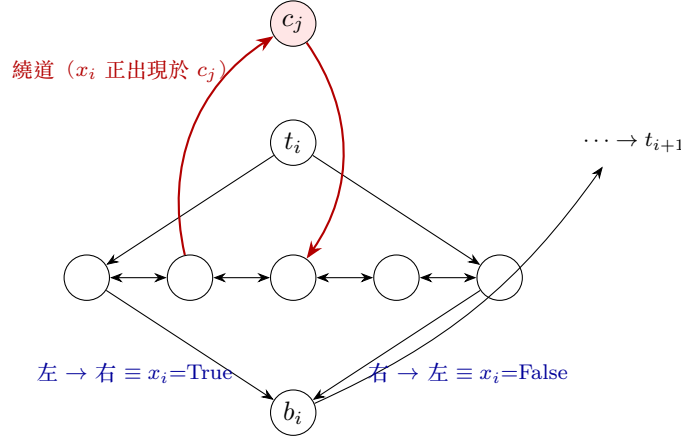
設 F 是 CNF 公式，變數為 $x_1, \dots, x_n$ 、子句為 $c_1, \dots, c_k$ 。標準建構 (「菱形鏈」構造)：

(一) 變數 gadget。為每個變數 x_i 建一個菱形 (diamond)：頂節點 t_i 、底節點 b_i ，中間夾一條水平鏈 $u_{i,1} \leftrightarrow u_{i,2} \leftrightarrow \dots \leftrightarrow u_{i,m}$ (相鄰節點以雙向邊相連)，並有邊 $t_i \rightarrow u_{i,1}$ 、 $t_i \rightarrow u_{i,m}$ 、

$u_{i,1} \rightarrow b_i, u_{i,m} \rightarrow b_i$ 。水平鏈可以由左往右或由右往左走——這兩種走法將分別對應 $x_i = \text{True}$ 與 $x_i = \text{False}$ 。鏈上為每個子句 c_j 保留一對相鄰節點（各對之間以分隔節點隔開，故 $m = O(k)$ ）。

（二）子句節點。為每個子句 c_j 建一個單獨節點 c_j 。若 x_i 以正文字出現在 c_j ：加邊「左節點 $\rightarrow c_j$ 」與「 $c_j \rightarrow$ 右節點」（只有由左往右經過時能順路繞去 c_j ）；若以負文字 $\neg x_i$ 出現：方向相反（只有由右往左時能繞去）。

（三）串接成環。 $b_i \rightarrow t_{i+1}$ ($i = 1, \dots, n-1$)，並加 $b_n \rightarrow t_1$ 使整體能閉合成迴路。



整體大小：節點 $O(nk)$ 、邊 $O(nk)$ ，建構顯然是輸入大小的多項式時間。取 G_F 為此圖，即得「 F 可滿足 $\iff G_F$ 有 Hamiltonian cycle」(由 (c)(d) 論證)。

(c) 可滿足 $\Rightarrow$ 有 Hamiltonian cycle (5 分)

設 σ 是 F 的滿足指派。建構迴路如下：

1. 從 t_1 出發，依序走過每個菱形：若 $\sigma(x_i) = \text{True}$ ，把第 i 條水平鏈由左往右走完；若為 False 則由右往左；走完從 b_i 接到 t_{i+1} ，最後由 b_n 回到 t_1 閉合。
2. 每個子句 c_j 因 σ 而含有至少一個為真的文字；挑選其中一個，設它來自變數 x_i 。若是正文字，則 x_i 的鏈正被由左往右走，恰好能在 c_j 的保留節點對處「繞道」：左節點 $\rightarrow c_j \rightarrow$ 右節點，再繼續前進；負文字則方向相反、同樣順路。每個子句節點恰繞道一次。

如此，所有菱形節點被主幹走過恰一次、每個子句節點被恰一次繞道拜訪，且首尾相接——這正是一條 Hamiltonian cycle。■

(d) 有 Hamiltonian cycle $\Rightarrow$ 可滿足 (5 分)

設 H 是 G_F 的任一條 Hamiltonian cycle。

（一）迴路必呈「正規形」。由建構，進入菱形 i 只能經 t_i 、離開只能經 b_i ；且子句節點 c_j 的出入邊都連在同一對相鄰鏈節點上，若迴路從某菱形繞去 c_j 卻從別的菱形離開，就會使該對節點中的另一個再也無法被拜訪（其僅存的鄰居都已用過），與 Hamiltonian 矛盾。因此 H 必定：逐一走過每個菱形，並把每條水平鏈完整地由左往右或由右往左走完，繞道後立即回到同一條鏈的下一個節點。

（二）由方向讀出指派。定義 $\sigma(x_i) = \text{True}$ 若 H 把第 i 條鏈由左往右走，否則 $\sigma(x_i) = \text{False}$ 。

（三）每個子句都被滿足。 H 是 Hamiltonian，必定拜訪每個子句節點 c_j ，而拜訪只能透過某個菱形 i 的繞道。繞道邊的方向設計使得：正文字的繞道只與「左往右」相容、負文字的繞道只與「右往左」相容。因此該繞道所在的文字在 σ 下為真，子句 c_j 被滿足。所有子句皆然，故 σ 滿足 F 。■

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