

【附件三】 成果報告（此為格式範例，詳情請見格式說明；請於系統端上傳 PDF 檔）

封面 Cover Page

教育部教學實踐研究計畫成果報告
Project Report for MOE Teaching Practice Research Program

計畫編號/Project Number：PEE1137340

學門專案分類/Division：工程

計畫年度：113 年度一年期 112 年度多年期

執行期間/Funding Period：2024.08.01 – 2025.07.31

在軟體程式課程中應用深度學習以協助題目設計和成效評估

計畫主持人(Principal Investigator)：李昌明

協同主持人(Co-Principal Investigator)：李詩偉

執行機構及系所(Institution/Department/Program)：中正大學/通訊工程學系

成果報告公開日期：立即公開 延後公開

繳交報告日期(Report Submission Date)：2025 年 9 月 16 日

本文與附件

計畫名稱：在軟體程式課程中應用深度學習以協助題目設計和成效評估

一、本文

1. 研究動機與目的

在主持人相關課程教學與辦理競賽過程中，已經逐漸掌握軟體程式在本系的定位，不同於資訊相關系所，系上學生需要的是能立足到硬體與軟體的領域發展，所以必須以 C 語言訓練為主，較少機會接觸高階程式語言(如 Python)，但後者因編譯器功能強大，不須嚴謹的程式架構即可完成同一題目，在近期校內外程式競賽屢屢限制系上學生的競賽成績，為尋求妥適的競賽與測試環境，需要對練習題目進行專業性與落地化的設計，所以需要建構能自行掌控的命題機制與自動驗題平台，考量資源與人力較不足，故欲引進深度學習技術作為彌補與強化，並尋求未來其它教學領域可援引的範例。然而，在一般的資訊科學教學的課程中，面臨的最大挑戰之一就是如何精確地控制作業以及考題的難易程度。傳統的出題方式往往受限於主觀判斷，導致難度評估容易受干擾。這不僅影響了出題的均衡性和適切性，甚至可能降低學習體驗。

為了解決這個問題，此計劃將引入先進且客觀的技術作為輔助。我們將採納使用深度學習的技術，如：ChatGPT[1]，作為出題和驗題的關鍵工具。ChatGPT 以其卓越的自然語言處理和生成能力而聞名，可以透過與其互動，動態生成各種具有挑戰性的題目，並以清晰而細緻的方式提供題解，為學習者提供更多的支援和引導。

這種整合深度學習的教學方式不僅使得出題過程更為靈活和客觀，也提供了一個更智能的解題環境，希望能夠根據學習者的表現，適當調整題目難度，確保每位學習者都能在挑戰中找到適切的平衡點。這種教學方式不僅僅是技術層面的升級，更是對於教學方法的一種創新。本計畫透過引入深度學習系統，希望能夠提供更具有挑戰性和實用性的學習內容，也為學習者打造了一個更豐富、深度參與的學習體驗。本計畫創新的教學方法將能夠提升資訊科技教育的水準，並為學習者提供更好的學習體驗。期望在落實這種教學方法後，能看到學習者在這種新的學習環境中取得的進步。

2. 研究問題

為了更深入地理解教師與助教如何利用深度學習系統來改進他們的設題技巧，並提高程式競賽或考試的質量，本研究主題將會針對三點問題做進一步的討論：

(一)教師與助教的設題思維和訓練成效：

研究者關注的是使用深度學習系統評估方法的有效性，可基於對話生成的評估方法根據教師與助教的設題思維和訓練成效，生成相應的對話回應，並給出評分和建議，這包括他們對題目難易度的把握、題目多樣性的設計、配合專業領域的素材以及解題策略的選擇依據。

(二) 深度學習系統評估結果與實際學生表現的一致性：

研究者將探討 ChatGPT 的評估結果與實際學生表現之間的一致性。如果存在差異，可能來自於教師與助教和學生的知識背景、理解程度、興趣偏好等因素導致。教師與助教可以根據這些差異，調整訓練方式，提高題目的適用性和有效性。

(三) 教師與助教根據深度學習系統評估結果並進行主觀評估的修正：

如果教師與助教的主觀評估和深度學習系統的客觀評估存在差異，研究者將探討教師與助教是否能夠靈活調整出題方式，進而提高題目的品質與難易精準度。例如，教師與助教可以做出增加或減少題目的難度、增加或減少題目的數量、增加或減少題目的類型、增加或減少題目的提示等調整。

這些問題意識將有助於研究者更深入地理解教師與助教如何利用深度學習系統來改進設題技巧，以提高程式競賽與考試的質量，同時也為學習者打造了一個更豐富、深度參與的學習體驗。這種智慧化的教學方式將成為本計畫的一大亮點，為學習者打開通向技術深度探索的大門。

3. 文獻探討

近期有相當多採用深度學習系統做為提升老師教學效率的討論，故認可深度學習系統具有一定的教學功能性，甚至訂定了一套生成式 AI 規範[2]，讓學生及老師能在合理的使用範圍內，提高學習及教學的品質與效率。目前時機成熟，可以利用深度學習系統作為命題的分級，減輕教師與助教的主觀意識，在群體測試時能夠更平均地分佈基礎級與進階級的題目數量，提高鑑別度及公平性，也同時兼顧到命題的有效性。

根據 Edgar Dale 的金字塔理論[3]，如果只有教師在課堂中上課的方式傳遞知識，學習效果只有 5% 至 30%，但如果在課堂中有實際演練或是教導他人，如此一來，學習效果可以提升到 50% 至 90%，而設計這樣的競賽能讓學生上機實作，賽後同學也能互相討論，甚至直接詢問出題人員更好的解題方式，提高學習效果，而不僅僅只是傳統的講述教學法。

透過這種競賽學習[4][5]的方式，能讓學生們達到做中學的方式，了解到自己需要補強的面向，也能在競賽中獲得一定的成就，提高學生的學習動機，避免挫折過大影響學習意願，裁判機系統也讓教師與助教能夠透過各位同學的解題狀況，藉由記錄分析以提供後續命題的調整，並針對學生的狀況做出更有彈性與精確的題目設計，而未來也有機會能夠與其它學習方法結合使用，例如專題導向式

學習[6][7]、團隊導向式學習[8][9]等等，對不同的資訊科技教育設計靈活的程式教學課程。

以往讓同學參加的 ITSA 程式競賽[10]屬於不分領域的程式競賽，本計畫為求能妥切應對專屬課程或專業領域，將自行架設裁判機伺服器[11]，並在專業領域以客製化方式命題及正確地驗算解決方法，能夠更彈性與自主地調整題目的難易程度，也可訓練出題人員擁有多樣化的思考模式，除了完善專業知識的學習，也能啟發學生具創造性的問題解決能力，以因應未來時代多變的需求[12]。

4. 教學設計與規劃

4-1 教學目標與方法

為打造資料驅動的教學迴圈，希望以競賽或上機測驗蒐集之「AI 互動資料」與實際成績為依據，滾動調整題庫與教學策略，所以本計畫的目標有三項：

(一)精準題目難度控制：

結合 ChatGPT 分析與教師審核，將題目分級，確保每位學生的表現落在近側發展區 (Zone of Proximal Development)。

(二)促進深度學習與反思：

以「AI 詢問次數」與「Prompt 精修紀錄」量化學生思考深度與修正能力。

(三)培養 AI 輔助開發素養：

於真實競賽場景體驗人機協作流程，依序進行題意解析、Prompt 撰寫、程式微調與正確解答 (Accept, AC)。

因此，本計畫將導入深度學習系統協助預判題目的難易度，而不只是使用教師與助教主觀的程度判別，深度學習系統評判的等第分為兩種：基礎級 (深度學習系統可以回答但解答的程式行數或巢狀迴圈數少)與進階級 (深度學習系統可以回答但程式行數或巢狀迴圈數多)，讓該次出題的題目整體難易程度不會全部過於簡單或是困難，理想狀況是儘量讓程式難易度在基礎級與進階級之間平均分配，使題目之間有鑑別度，也能更好看出同學在連續測試時的程度變化。

4-2 課程設計與規劃

本計畫將搭配兩門課程，雖有屬性與內容的差異，仍會依照課程的進度與學生的學習狀況進行彈性調整。

(一)資料結構：

此課程屬於大二必修課，課程安排須涵蓋幾項必須的授課主題，如 Pointer、Stack、Linked list……，學生必須理解並能描述資料結構的基本概念，包括數據類型、數據表示和數據處理，教師與助教必須確保適宜的進度完成各項學

習，各項課程活動（如作業、小考、上機練習與期中考、期末考）必須確保各單元的均衡分配，故課程進度較為穩定。

(二) 軟體程式開發實務與實作：

此課程屬於大三的選修課，除補充部分演算法的知識，多數採個案研究的方式，先透過數次的上機測試與競賽累積答題經驗，接著便嘗試解題與操作深度學習系統完善整體測試的難易度，以預估與回饋方式鍛鍊解題技巧，需要保持較高的自主學習動機，以增加 Prompt 撰寫的靈活度與精準度。

4-3 學生成績考核與學習成效評量工具

(一) 資料結構：

因"資料結構"屬於大二必修課，需要確保學生能完整學習，必須以傳統上機測驗與筆試完成成效評量，但本計畫會提供整體均衡的題本，學生可以逐次比對每次測驗的成效是否能反映學習的效益，並養成自我檢核與除錯習慣。評量方式會由學生成績呈現四次上機測驗每次佔總成績 15% 總計 60% 上機測驗每次皆有兩題 第一題是單純的資料結構實作題 檢測學生是否有能力實作出指定資料結構佔該次上機測驗 80 分，剩餘 20 分為資料結構應用題，會有較大的變化以及包裝，考驗學生對資料結構更深入的了解以及應用。成效評量則呈現在測驗補測機制中，學生參與 Prompt 工坊 (Prompt Engineering Workshop) 時，會在測驗後對題目較為了解的狀況下，以 LLM 輔助解題，記錄其提示與解題過程（最多五次）。另有兩次筆試各佔總成績 20%，考驗學生對各種資料結構的時間複雜度以及證明等理論的熟悉度。

(二) 軟體程式開發實務與實作：

為能全面評估學生在學習的成效，"軟體程式開發實務與實作"將採用多元的學生成績考核和學習成效評量工具，主要的評量方式將是透過舉辦程式競賽，以呈現學生在實際應用中所獲得的技能和知識。本課程將設計一系列挑戰性的程式競賽題目，充分反應程式設計的不同層次和面向。這些題目將涵蓋許多演算法、資料結構和問題解決方法，以確保學生在競賽中需要運用多方面的知識。題目的難易度將由教師或助教事先評估，以確保競賽的公正性和挑戰性，參與競賽的學生將根據他們在程式競賽中的表現獲得評分，以全面評價學生的程式能力。評量方式會由學生兩次線上測試各佔總分 25 分，每次 100 分支配分，計分公式為

$$10*f + (a+a'/2)*40/q \quad (1)$$

其中 f 為每題解題之流程圖 (flowchart, 最多 6 題), a 為線上測試 AC 的題數, a' 為線上測試 NA 但在 prompt 工坊 以 copilot 輔助補測有通過 (至多 2 題), 線上測試與輔助測試總和至多 6 題, q 為線上測試 AC 的合格題數。接下來尚有總分 50 分針對線上題庫 (LeetCode 與 ITSA) 作業。學生須

繳交 10 題之提示與解題過程 (每題最多五次，若 5 次仍不通過則以人工分析關鍵並介入重寫)，其中難易最多各 5 題。成效評量主要會以上述的 Prompt 工坊的項目呈現，包涵測驗後的補測與線上題庫之作業。

4-4 教學流程 (學期節點)

(一) 資料結構課程安排：

因本課程屬於大二必修課，將教導學生在電腦系統做到高效儲存、整理與處理資料，從而優化資料存取的速度。課程涵蓋陣列、堆疊、佇列、鏈結清單、樹狀結構及圖形等基本資料結構的原理、實作方法與效能分析，目的是幫助學習者掌握資料處理的基礎機制，並為後續的程式撰寫、演算法開發以及軟體工程打下穩固根基。

表 1: 資料結構課程安排

第 1-4 週	資料結構簡介以及 C 語言指標介紹
第 5-6 週	Stack 資料結構教學以及應用
第 7-8 週	Queue 資料結構教學以及應用
第 9 週	期中考筆試
第 10 週	Linked list 資料結構教學以及應用
第 11 週	Linked list 資料結構及應用上機測驗及 <u>Prompt 工坊補測</u>
第 12 週	Binary tree 資料結構教學以及應用
第 13 週	Binary tree 資料結構及應用上機測驗及 <u>Prompt 工坊補測</u>
第 14 週	Heap 資料結構以及 Quicksort 演算法教學
第 15 週	Heap 資料結構以、Quicksort 演算法上機測驗及 <u>Prompt 工坊補測</u>
第 16 週	圖論介紹以及最小生成樹 最短路徑等演算法教學
第 17 週	圖資料結構上機測驗及學生 <u>Prompt 工坊補測</u>
第 18 週	期末考筆試

(二) 軟體程式開發實務與實作課程安排：

本課程安排聚焦程式競賽與 AI 解題，先介紹 Domjudge [11] 架設，接著進行熱身賽、正式賽與練習賽，包括測試、賽後補測及系級競賽。並融入線上 LeetCode/ITSA 演練，業界分享。

表 2: 軟體程式開發實務與實作課程安排

第 1-2 週	程式競賽裁判機 (Domjudge) 介紹與架設示範
第 3 週	熱身賽 (班內第一次測試)，離線測試與 AI 解題測試
第 4 週	<u>Prompt 工坊</u> ：熱身賽賽後補測
第 5 週	正式賽 (班內第二次測試)，系級競賽與 AI 解題測試
第 6 週	<u>Prompt 工坊</u> ：正式賽賽後補測

第 7-10 週	線上程式庫(LeetCode/ITSA) 線上 AI 解題演練與業界教師經驗分享 a. 楊朝棟經理建興儲存科技 b. 陳雋尹博士 斜槓創作者
第 11 週	練習賽 (班內第三次測試)，離線測試與 AI 解題測試
第 12 週	<u>Prompt 工坊</u> ：練習賽賽後補測學生
第 13-18 週	線上程式庫(LeetCode/ITSA) 線上 AI 解題演練

4-5 教學方法與工具

(一) 資料結構

a. 結合傳統教學與實作導向學習：

教師採用結構化教學方式，透過板書、講解與範例程式，系統性介紹資料結構的核心概念 (如 Stack、Queue、Linked List、Binary Tree 等) 及其演算法應用。為加深學生理解，每單元搭配精心設計的段落式練習題，涵蓋基本操作與進階應用，引導學生逐步掌握資料結構的實作邏輯與理論基礎。

b. 筆試與上機測驗並重：

課程評量結合筆試與上機測驗，全面檢視學生學習成效。筆試聚焦於資料結構的理論理解，例如時間複雜度分析與演算法證明，確保學生掌握觀念基礎。上機測驗則強調程式實作與除錯能力，透過真實編碼情境檢驗學生將理論轉化為實務的能力，實現學用合一的目標。

c. Prompt 工坊與補測機制：

每次上機測驗後，設置「補題挑戰」環節，學生在限定次數 (最多五次)內與 ChatGPT 協作解題，目標是生成通過測試 (AC) 的程式。過程中，學生需記錄與 ChatGPT 的互動歷程，包括 Prompt 的撰寫與調整，培養精準提問與自我反思能力。此機制不僅促進學生對資料結構的深入理解，還鼓勵其探索 AI 輔助解題的策略，提升程式設計的靈活性和效率。

d. 程式輔助教學工具：

課程導入 Domjudge 裁判機系統，作為上機測驗與補測的平台，記錄學生提交的程式碼與執行結果，提供即時回饋，幫助學生檢視錯誤並優化程式。

(二) 軟體程式開發實務與實作

a. AI 出題／驗題雙層輔助機制：

教師與助教負責生成題目敘述、限制條件與範例測資，確保題目符合課程目標與專業需求。ChatGPT 則作為輔助工具，複核題目的語意歧義、邏輯正確性與難度分布，幫助優化題目品質並維持均衡的挑戰性。

b. Prompt 工坊：

透過專門設計的 Prompt 工坊，培養學生有效利用 AI 進行解題的能力。重點包括：

- (1) 語句格式：學生學習以「操作式說明」向 AI 明確指示問題需求，提升溝通精準度。
- (2) 結構語彙：學生逐步熟練使用資料結構或演算法相關的專有名詞作為提示核心，強化專業表達。
- (3) 限制條件：學生有意識地加入「問題環境約束條件」，以控制 AI 生成內容的品質與適用性。學生限定在 5 次內讓 ChatGPT 生成通過測資 (AC) 的程式，並比較不同 Prompt 寫法的成效，藉此訓練反覆思考與問題解決能力。
- (4) 成功回應傾向：強調 Prompt 的明確性對 AI 產出結果的決定性影響，引導學生優化提問策略。

c. AI 互動資料反饋：

自行建置查詢系統，記錄學生與 AI 的提問、回覆、程式提交與執行結果。每週彙整「詢問次數分布」與「AC 率」，提供教師與助教作為出題與教學調整的依據，實現資料驅動的教學優化。

5. 研究設計與執行方法 (Research Methodology)

5-1 教學方法與工具

本計畫是利用深度學習系統的輔助來評估程式競賽題目的難度，並以此提升教師與助教的命題品質。教師與助教會根據自己的經驗設計題目，以減少蒐集大量題目以及對應難度等級資料庫的時間，再使用深度學習系統來評估這些題目的難度和可能的解法。然後，他們會比較參賽者的實際表現與深度學習系統的評估，再與教師與助教的主觀評估進行驗證。最後，他們會根據反饋來調整教師與助教後續出題的難度。這種方法有助於提高題目設計的準確性，並提供更好的學習經驗給參賽者，整體流程如下圖 1 所示。

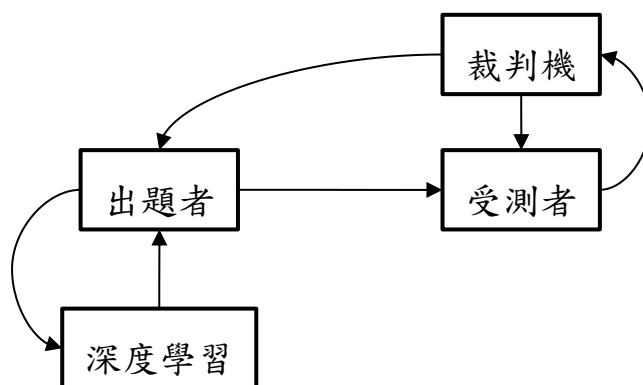


圖 1. 深度學習系統輔助題目設計之計畫架構

5-2 研究範圍目標

本計畫透過個案研究的方式，深入探討深度學習系統於每次測驗設計题目的技巧，並透過此方式來訓練出題人員，在不同的年級和課程中測試適用性以及有效性。這個目標關注的是深度學習系統的評估方法是否能夠適應不同的題目類型、難度等級和學習目標，並對教師、助教及學生產生正面的影響。再進一步可評估循環式訓練對教師與助教的影響，包括是否能夠快速修正命題方式、提高主觀評估的準確性，進而提升题目的品質。

另外，也將探討結合不同的評估方法來提升程式競賽的品質，這個目標關注的是深度學習的評估方法與其他評估方法的區別和優勢，以及如何利用不同的評估方法的互補性和協同性來提升程式競賽的設計、執行和評估。

5-3 研究對象與場域

研究對象為有兩類，第一類為具有一定經驗的程式競賽教師與助教，教師與助教的專業經驗對於評估深度學習系統的應用和訓練效果至關重要。他們的反饋和調整將有助於研究更有效的訓練策略。第二類為受測對象，主要是大學部三年級與二年級學生，以 113 學年度上學期針對大學部學生所開授 "軟體程式開發實務與實作"與"資料結構"的兩門課程為研究場域，如圖二所示，這兩門課是提供常規評估學習成就的關鍵資料，他們的回饋和表現能夠反應深度學習系統在程式教學課程中的實際效果。接著再將研究場域擴展到全系各年級同學，搭配辦理電機系與通訊系的程式競賽，檢驗全面性的成效。

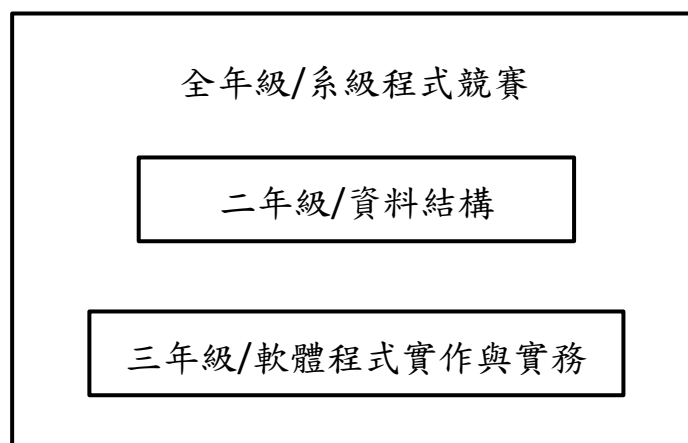


圖 2. 受測身分與課程活動範圍

5-4 研究方法與工具

本研究以個案研究為主要研究方法，以深度學習系統 (如 ChatGPT) 作為評估題目難度、評估解題策略之工具，用以輔助教師與助教的命題過程，另外架設 Domjudge[11]為程式競賽的平台，讓學生在實際競賽中進行解題，根據裁判機紀

錄之競賽成績（包含成功答題比例與正確率）和出題人員評估的答題情況來做為修正出題方向的依據。

5-5 研究實施程序

首先，由教師與助教完成命題，先經由教師與助教對自己所出題目的認知做主觀的難易度評斷，再將題目提供給深度學習系統評斷難度，可分為：基礎級（深度學習系統可直接提供最優解）與進階級（深度學習系統需要進一步提示才能提供解答或不能符合限制的解答），若與教師、助教主觀認定的難易度不同，將再調整題目的陳述細節，最後經由學生的答題成績以及情況來驗證此題的難度是否如教師與助教所預期，可在資料結構以及軟體程式開發實務與實作的課程各進行至多次的實驗，評估後再以系級競賽的方式將受測對象擴展至全年級的同學，搭配適當的難易程度題型的比例。以下是該研究架構的詳細步驟：

步驟一、教師與助教設計題目：教師與助教將根據其出題經驗來設計特定難度的題目。這些題目將涵蓋不同的主題和難易度。

步驟二、使用深度學習系統評估題目：教師與助教在完成題目設計後，將利用深度學習系統擔任第三方的角色來評估題目的難度和可能的解法。

步驟三、評估參賽者的表現：研究人員將評估參賽者在競賽中的表現，其中包括正確率、解題速度等指標。

步驟四、比較與分析：研究人員將參賽者的實際表現與深度學習系統的評估進行比較與分析。

步驟五、驗證與修正：研究人員將比較與分析的結果和教師與助教的主觀評估進行驗證，並根據這些反饋修正教師與助教後續出題的難度控管。

以下提供兩級題型的範例與判定：

a. 基礎級範例

題目：給定數字 n ，輸出由 $1 \sim n$ 的總和

ChatGPT 回應：可直接輸出了時間以及空間複雜度都是最優解的方法，以及同時給定正確的程式碼。

b. 進階級範例

題目：給定數字 n 以及 k ，代表接下來有一個包含 n 個數字的陣列以及 k 次操作，每次操作可以將當前陣列中任意兩個數字相減後取絕對值並插入回陣列中，求經過 k 次操作後陣列中的最小值，時間複雜度最高為 $O(n^3)$ 。

ChatGPT 回應：首先，在理解題目方面就有明顯的問題，ChatGPT 無法考量兩個附加條件，他的程式碼去考慮 k 的狀況，本題有關鍵的算法可以忽略 k ，但 ChatGPT 在最初並無法第一時間看出這點並給予正確回應，並且在多次提示後仍無法給出正確解答，甚至在不斷修正的過程中，從原本只是時間複雜度過高，修正到連整個程式的輸出都是錯誤的。

6. 教學暨研究成果 (Teaching and Research Outcomes)

6-1 教學過程與成果

(一) 資料結構：

第一次 Prompt 工坊 (第二次上機測驗)：第一題平均 1.44 次，第二題 1.58 次；總完成率約 85%。

第二次 Prompt 工坊 (第三次上機測驗)：題目加深，平均提示提升至 1.71 與 1.77 次；完成率約 80%。

第三次 Prompt 工坊 (第四次上機測驗)：學生回穩，平均提示降至 1.52 與 1.32 次，第二題完成率高達 93%。

每次都有超過三十位學生進行測試，可以看到學生在反覆練習中，能把 Prompt 精煉到 1-2 次即拿到 AC，達成「低嘗試、高正確」的學習目標。

(二) 軟體程式開發實務與實作

第一次測試 (表 3) 的平均答對率約 72.17%，顯示題目相對基礎且均衡，大多數題目有較高通過率，如(D)作家 91.67%、(B)/(F)深海/基因 83.33%，符合計畫中“基礎級”題型的特徵 (ChatGPT 易直接提供最優解)。這反映出首輪作為熱身賽的設計意圖，旨在讓學生建立信心並熟悉系統。

第二次測試 (表 4) 的平均答對率驟降至約 36.83%，題目難度明顯提升，例如(E)校慶煙火 0%、(H)漢明距離 1.79%、(A)熊派對 10.71%，這與報告中提到的“題目難度提高”一致。少數題目維持高答對率，如(C)哥布林 98.21%，可能為基礎錨定題，用以平衡整體鑑別度。這種難度梯度設計支持計畫目標：確保題目落在近側發展區，避免全簡單或全困難，同時提升鑑別度以追蹤學生進步。

出題者提示次數用來預判難度 (高次數表示進階級，ChatGPT 需多輪提示或無法正確解題)。表 3 平均約 2.4 次，表 4 平均約 2.8 次，顯示第二次測試題目更複雜 (如(A)熊派對 4 次、(H)漢明距離 3 次)。引人注意的是，這與學生答對率負相關：出題者提示多的題目，如第一次測試的(A) 題，學生答對率低。這驗證了計畫研究方法 (深度學習系統作為客觀輔助) 的有效性，能幫助教師/助教預估並調整難度，減少主觀偏差。如報告中進階級範例，當給定 n 和 k 的操作題提到 ChatGPT 初次無法正確回應，與表 4 中高提示題的特性類似。

再針對參賽答對者平均上傳次數做分析，表 3 平均約 2.25 次，表 4 平均約 2.84 次，表明第二次測試需要更多反覆除錯。這反映學生面對難題時的堅持度增加，例如第二次測試的 (A)熊派對 4.17 次、(D)搶救普物 3.54 次，相比表第一次測試的 (D)作家 4.36 次 (雖高但答對率高，可能是細節陷阱導致)。根據彙整的數據 (表 3: 第一次測試平均答對率 72.22%; 表 4: 第二次測試平均答對率 36.83%)，可以比較兩次測試的答對率(如圖 3)，顯示兩次測試的難度梯度變化，從第一次的較低難度到第二次的較高難度。同時，進行獨立樣本 t 檢驗來確認差異顯著性

($p=0.04$)。這數據顯示過高難度可能影響動機，但透過 Prompt 工坊的補測機制，能轉化為學習機會，培養程式微調與正確解答的能力。

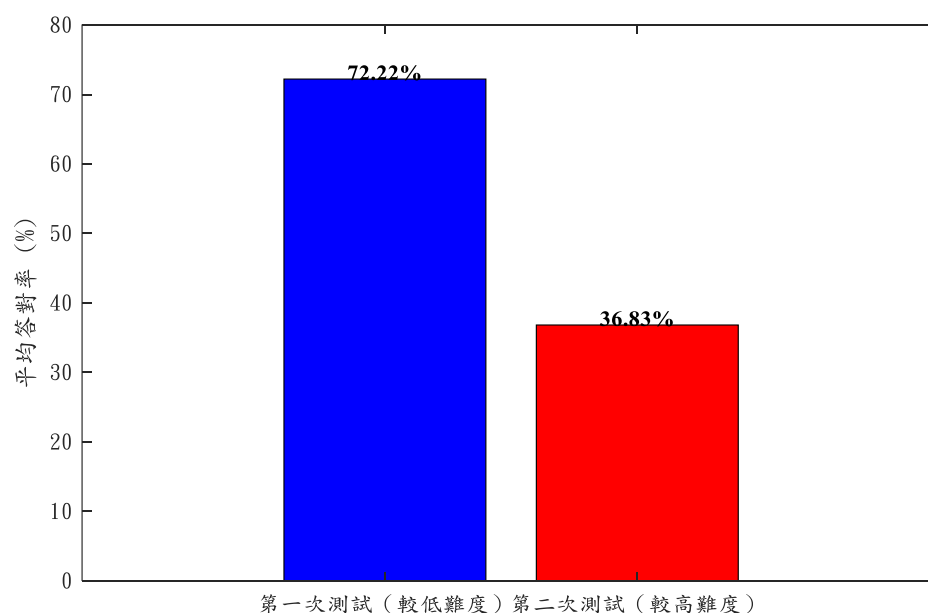


圖 3. 以答對率比較顯示難度梯度差異

再透過 Prompt 工坊 ChatGPT 提示次數量化學生 AI 互動深度，是本計畫的亮點，表 3 平均約 1.58 次，表 4 平均約 1.89 次，雖略升但仍維持在低水平 (最多 5 次限制內)，顯示學生逐漸掌握 Prompt 工程技巧。例如，第二次測試的 (E) 校慶煙火 3 次 (答對率 0%，需更多提示)，但整體趨勢是(精準 Prompt → 快速 AC)，如報告所述，從初次模糊提問進化到使用專業詞彙 (如時間複雜度)。這證明 AI 輔助不僅彌補難度差距，還促進反思：學生記錄互動歷程，能量化思考深度，達成計畫目標“促進深度學習與反思”。

表 3: 第一次上線測試與提示統計

題目	出題者使用 ChatGPT 驗題所需提示次數	參賽者答對率	參賽答對者上傳裁判機平均次數	Prompt 工坊使用 ChatGPT 所需提示次數
(A) 跳島戰術	NA	58.33%	3	1.86
(B) 深海研究計畫	5	83.33%	1.2	1.57
(C) 塗色問題	2	50%	1.5	1.43
(D) 作家	2.5	91.67%	4.36	1.43
(E) ID 生成器	1	66.67%	1.125	1.67
(F) 基因	1.5	83.33%	2.3	1.5
平均	2.4*	72.17%	2.25	1.58

*：排除 NA

表 4:第二次上線測試與提示統計

題目	出題者使用 ChatGPT 驗題所需提示次數	參賽者答對率	參賽答對者上傳裁判機平均次數	Prompt 工坊使用 ChatGPT 所需提示次數
(A) 熊派對	4	10.71%	4.17	1.17
(B) 採收鳳梨	NA	37.5%	3.29	2
(C) 哥布林	2	98.21%	1.24	1.33
(D) 搶救普物	NA	50%	3.54	2.33
(E) 校慶煙火	NA	0%	NA	3
(F) 裝置藝術	3	69.64%	2.82	1.67
(G) 地圖縮放	2	26.79%	1.80	1.25
(H) 漢明距離	3	1.79%	3.00	2.4
平均	2.8	36.83%	2.84	1.89

*：排除 NA

以上的分析已經呈現難度提升導致答對率與效率指標下降，但 Prompt 工坊提示次數僅微增，顯示學生適應力強。這與報告 6-1 節的總結一致：學生在反覆練習中達成低嘗試與高正確的表現，並在表 5 與表 6 (LeetCode/ITSA) 實驗中維持類似水平 (基礎題 2.38-2.17 次)。數據也反映循環調整的有效：教師根據 AI 評估與學生表現滾動優化題庫，實現資料驅動教學。兩表數據顯示學生在線上題庫中維持高效 AI 互動，平均提示次數低於班內測試 (表 3 與表 4 約 1.7 次)，反映計畫的滾動調整成效：以“AI 詢問次數”量化思考深度，結合 Domjudge 裁判機記錄，優化題庫與教學策略。可達到培養 AI 輔助開發素養的目標，並與 Edgar Dale 金字塔理論一致——透過實際演練 (競賽+AI 補測)，學習效果提升至 50-90%。再細究可察覺 LeetCode 進階提示較高，可能因全球題庫多樣性，建議未來擴大人工審核 (如報告 7-1 反思：AI 需反覆驗證)。ITSA 進階低提示或許過於樂觀，若數據偏差，需探討學生背景因素。

表 5: LeetCode 上線測試與提示統計

題目難度	學生平均正確解題所需提示次數
普通	38 次/16 題=2.38
進階	29 次/11 題=2.64

表 6:ITSA 上線測試與提示統計

題目難度	學生平均正確解題所需提示次數
普通	156 次/72 題=2.17
進階	69 次/34 題=2.03

6-2 學生學習回饋

(一) 資料結構

教學意見平均 4.69 (滿分 5 分)，學生意見陳述如下：

1. 我覺得老師講解觀念的時候都非常清楚，但因為我自身的程式能力不足，需要更多的自我練習。非常感謝老師這學期的教導。
2. 整體很棒！
3. 我喜歡老師上課的節奏，老師也不會太常跳過不講課程內容，叫我們回家自己讀。
4. 老師的考試安排對大家來說，只要肯唸就有 80 分，非常感謝老師，但還是有人要作弊。
5. 老師很認真。
6. 教學內容扎實，從這門課程學了很多東西。
7. 我覺得這堂課唯一的缺點就是筆試，考的內容很細，要寫到很完整才有分，如果筆試能改成填空或選擇，我相信會更好，但那是學弟妹的事了，可以維持筆試。
8. 老師教學用心，詳細講解課程內容。

(二) 軟體程式開發實務與實作

此課程為電機與通訊併班，但教學意見調查是分開。

a. 電機班: 教學意見平均 4.29 (滿分 5 分)，學生意見陳述如下：

1. AI 很笨
2. 老師人很好，作為交換學生來到台灣，感受到了台灣老師的關心和熱情，遇到問題老師還會主動關心你的非常負責任老師還會請學長學姐分享經驗，課程結束還請我們吃好吃的食物，總的來說，謝謝老師，很推薦這門課程。

b. 通訊班: 教學意見平均 4.12 (滿分 5 分)，學生意見陳述如下：

1. 作業稍微有點麻煩 二、三次考試題目有點難，感覺第一次測試就定生死了。

7. 建議與省思 (Recommendations and Reflections)

7-1 教師教學反思

(一) AI 輔助需人工複核以確保品質：

在首次測試中，發現 Copilot 生成的程式碼偶爾存在邏輯錯誤，若未經教師審核，學生可能誤用不正確的解答。因此，後續採取「AI 生成 → 助教審核 → 線上裁判機驗證」的三階段流程，顯著降低錯誤率，確保題目與解答的正確性與可靠性。進一步反思，AI 工具如 ChatGPT 在生成程式碼時不能直接解決問題，剩餘部分仍需人力介入以處理複雜邏輯或特定情境，這強調了教師在 AI 輔助流程中的關鍵角色，避免過度依賴導致學生誤學。

(二)題目設計應循序漸進並兼顧挑戰性：

第二次測試因題目難度提升，學生所需提示次數增加，顯示過高的挑戰可能影響學習動機。為此，後續調整題目設計為「核心概念 + 單一陷阱測資」，既能檢驗學生對資料結構與演算法的理解，又避免過度挫折，維持學習的成就感與可達成性。為防止 AI 輔助可能減少認知負荷，需要加強學生更紀律的思考；因此，教師應設計漸進式題目，從基礎 MFU (Minimal Functional Units) 出發，讓學生適應 AI 生成後的修改，而非從零開始，以提升效率並減少挫敗。

(三)提前進行 Prompt 撰寫訓練：

初期學生常使用模糊的提問，如「請幫我寫程式」，導致 AI 回應效率低。經過 Prompt 工坊的系統性訓練，學生學會使用精準關鍵詞，如「 $O(N \log N)$ 、處理負權邊」，顯著提升與 AI 互動的效率，縮短解題時間並優化程式成果。經檢視發現結構化提示不僅改善互動，還培養學生批判性思考；例如，鼓勵多步驟提示（如分析特徵、比較需求、修改代碼），這與研究一致，能顯著提高學生自信並減少除錯需求。

7-2 建議與未來研究方向

未來研究將聚焦於程式題目設計與測驗的雙向參與者，涵蓋教學方（教師與助教）及學習方（學生），並設定以下兩大目標：

(一)提升教學成效：

透過線上深度學習工具輔助，生成難度均衡且適切的題目，協助教師與助教更有效檢視學生學習成果，同時縮短命題時間。這種方式不僅減輕教學負擔，還能強化學生獨立解題與自我檢核的能力，提升整體教學效率。建議教師設計 MFU 作為技術框架，讓學生從適應現有邏輯語法開始，搭配結構化提示策略，以優化 AI 整合並轉變教師角色為合作協調者。

(二)深化學習體驗：

利用深度學習系統的互動性與智能性，提供清晰且詳盡的題解，引導學生多元理解複雜演算法與資料結構概念，培養自主思維與問題解決能力。透過形成性評量 (Formative Assessment)，持續蒐集學生在競賽與測驗中的表現數據，動態調整教學策略與出題方向，確保題目難度與課程目標一致，為學生打造更具挑戰性與實用性的學習內容。建議整合 AI 配對程式設計作為補充，特別針對焦慮或動機低的學生，以填空或選擇題型協助學生建立信心，以減少壓力並提升表現，但需平衡與傳統人類配對以維持合作技能發展。

未來可擴大研究範圍，探討 AI 在程式教育中的倫理議題，如公平存取、學術誠信及潛在偏差。建議開發全面策略，包括教師訓練及學生 AI 素養培養，以確保廣泛採用。基於文獻[13]，可探索 AI 工具在軟體開發生命週期中的應用，從需求分析到維護與除役的協同與互補，作為新的軟體工程訓練型態。同時，進行更大規模研究以驗證 MFU 框架的通用性，擴展到不同年級與背景。

此計畫另一衍生的發展，可進一步探索如何整合深度學習系統的動態反饋功能，為教學方與學習方提供更多支援，激發學習興趣與動機，創造豐富且深度參與的學習體驗。同時，開發更精準的評量工具，透過數據分析評估課程成效，推動智慧化教學模式的持續優化，為學習者開啟技術探索的新路徑。

二、參考文獻 (References)

- [1] ChatGPT, <https://chat.openai.com>, Access time September 16, 2025.
- [2] 臺大針對生成式 AI 工具之教學因應措施, <https://www.dlc.ntu.edu.tw/ai-tools/>, Access time Dec. 13, 2023.
- [3] K. Masters, "Edgar Dale's pyramid of learning in medical education: a literature review," *Med Teach.* 2013;35(11): e1584–e1593. doi:10.3109/0142159X.2013.800636
- [4] 王怡又, 黃義良, 一所科技大學競賽團隊學生學習歷程之探究, DOI: 10.7038/JETP.202012_(42).0001
- [5] J. C. Burguillo, "Using game theory and competition-based learning to stimulate student motivation and performance," *Computers & Education*, 55(2), pp. 566-575, Sept. 2010.
- [6] H. S. Barrows, "The tutorial process," SIU School of Medicine, 1992.
- [7] D. Moursund, "Project-based learning using information technology," Eugene, OR: International Society for Technology in Education International Society for Technology in Education. 2002.
- [8] 以學生學習為中心的教學：團隊導向學習法, <https://teaching.ndhu.edu.tw/files/15-1095-41700,c8448-1.php>, Access time Dec. 13, 2023.
- [9] 巫俊采, "結合團隊導向學習與實務學習體驗之教學實踐研究," *教學實踐與創新*,5(2), pp. 53-110, Sept. 2022.
- [10] 程式自學平臺, <https://e-tutor.itsa.org.tw/e-Tutor/itsaojcontest/index.php>, Access time Dec. 13, 2023.
- [11] DOMjudge - Programming Contest Jury System, <https://www.domjudge.org/>, Access time Dec. 13, 2023.
- [12] V. Garbuz and P. Topala, "The trend of higher engineering education towards achieving technological development," IOP Conference Series Materials Science and Engineering, August 2019.
- [13] Tom Taulli, "AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment," O'Reilly Media, 2024.

三、附件 (Appendix)

本附件為本計畫成果報告之補充資料，記錄 113 學年度電機通訊軟體競賽活動。該競賽由中正大學通訊工程學系主辦，場地設於創新大樓 504 及 505 室，旨在透過程式競賽驗證深度學習輔助題目設計之成效。



電機通訊程式競賽 113

Problem A
跳島戰術

Time limit: 1 second
Memory limit: 256 megabytes

題目內容

美國五星上將麥克阿瑟曾有多句名言，也在整個生涯對於太平洋海站有許多戰略，其中在二戰末期執行了一個名為跳島戰術的作戰計劃，其核心是選擇性攻占戰略位置重要的島嶼，而非每一塊敵佔領的土地。

現有 n 個島嶼，每個島嶼有各自的戰略價值 val_i ，在這個戰術裡，佔領的島嶼不能兩兩相鄰，單單想要獲得最高的總戰略價值。

其中總戰略價值的定義為：佔領的島嶼中的**最高**戰略價值 + 佔領的島嶼數量

請輸出一個數字，代表在這個戰術下，經過頑密的計畫後，可能達出的最高總戰略價值。



輸入格式

輸入共有兩行。

第一行有一個數字 n ，代表島嶼的總數量。

第二行有 n 個數字 val_i ，為第 i 個島嶼的戰略價值。

輸出格式

輸出一個數字，為能獲得的最高總戰略價值。

技術規格


- $1 \leq n \leq 2 \times 10^5$
- $1 \leq val_i \leq 10^6$

範例輸入 1 **範例輸出 1**

3	7
5 4 5	


1


(a)



電機通訊程式競賽 113

Problem E
校慶煙火

Time limit: 1 second
Memory limit: 256 megabytes

題目內容

每年中正大學都會舉辦校慶煙火，吸引無數校友和學生回來觀賞。然而，這一年校方決定要舉辦一場史無前例的華麗煙火秀，讓學生們感覺到所做學費的價值不只是上課學習，還能享受到一場視覺盛宴。然而，煙火的點燃方式可不是那麼簡單，這一次的煙火秀加入了一個遊戲挑戰。

中正大學的煙火裝置共設有 n 個點火器，代表著一組特殊的煙火效果。這些點火器一開始都處於未點燃狀態，與此同時，有 n 個按鈕可以控制這些點火器。按下第 n 個按鈕時，所有編號為 n 的倍數的點火器將會被點燃或熄滅（即狀態反轉）。

同時，為了讓煙火秀更加有趣，設計了一些規則：

- 你至少需要按下一個按鈕；
- 每個按鈕只能按一次；
- 有 m 對規定 (u_i, v_i) ，如果你按下按鈕 u_i ，那麼你必須按下按鈕 v_i （按下的順序不限）。但如果你按下了按鈕 v_i ，那麼你就不必再按下按鈕 u_i （可按可不按）。

你的任務是找到一種按按鈕的方式，讓煙火的最終點燃效果不會過度繁瑣——最多只能有 $\lfloor \sqrt{n} \rfloor$ 個點火器被點燃，否則煙火秀將顯得過於混亂。若無法達成這個條件，則只能遺憾地宣布挑戰失敗，無法放出完美的煙火秀。

希望每個校友和學生都能見證這場最為精緻的校慶煙火秀，你能夠完成這個挑戰，讓這場校慶煙火成為校史上最閃耀的一刻嗎？

輸入格式

第一行有一個數字 n ，代表會有 n 筆測資



每筆測試中的第一行分別有兩個數字 n, m 中間以空白隔開，分別代表 n 個點火器以及 m 條規則

接下來 m 行，每行有兩個數字 u_i, v_i 為按鈕順序之規則

輸出格式

如果無法完成挑戰，輸出 -1

否則，首先輸出一個整數 k ，表示按下的按鈕數量。接著輸出 k 個按鈕的編號，編號必須是不同的，且最終點燃的點火器數量不超過 $\lfloor \sqrt{n} \rfloor$ 個


9


(b)

附件圖 1. 題目範例 (a)班內(第一次)測試基礎級 A 題 (b)系競賽(第二次)測試進階級 E 題



附件圖 2. 比賽場地全景，參賽學生於斷絕外部網路電腦編寫程式，裁判機 Domjudge 系統運作中。



附件圖 3.主任頒發獎項予獲勝團隊，鼓勵學生程式素養之優異成果。



附件圖 4. 贊助企業代表參與頒獎，象徵業界連結，連結計畫未來方向。