

LLM 使用比較與心得

吳瑞宸 / s1143338 / 資一 C / 2025-12-23

三個模型名稱

- 模型 A :

Google Gemini / <https://gemini.google.com/app?hl=zh-TW> / 2025-12-23

- 模型 B :

ChatGPT / <https://chatgpt.com> / 2025-12-23

- 模型 C :

Perplexity AI / <https://www.perplexity.ai> / 2025-12-23

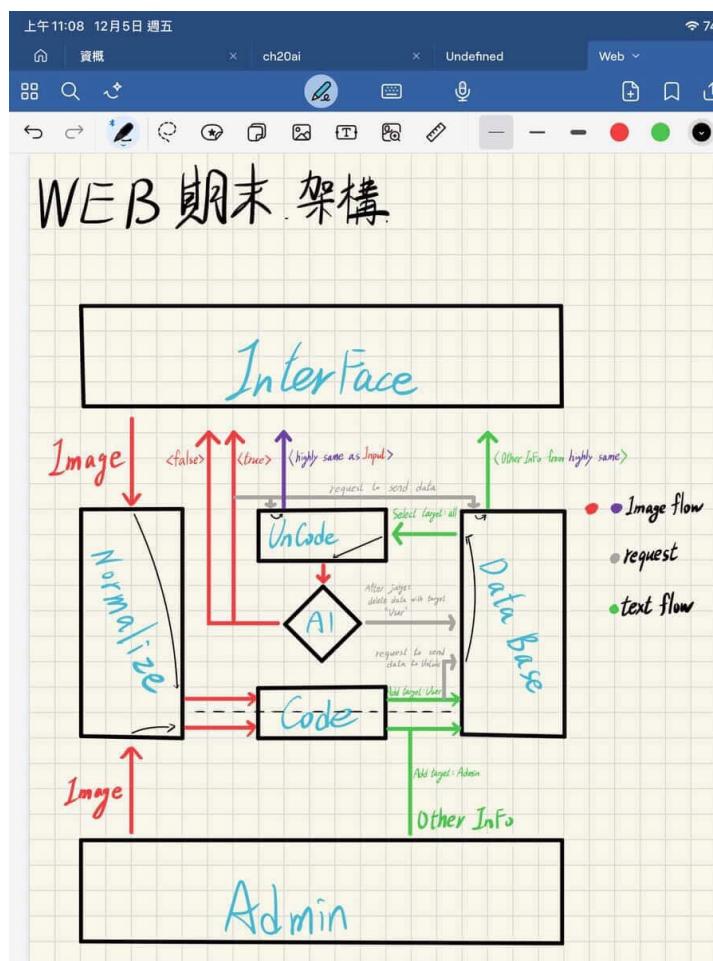
1. 任務一：優化 Web 期末專題的架構

- 我的 Prompt :

”

我是資訊工程學系大一的學生，這是我的 Web 程式設計期末專題的架構圖，我想要完成一個可以透過前端接輸入的圖片，與資料庫(資料庫中的照片為管理員所新增)(透過 Base64 編碼儲存)，具一定程度上的統一格式)中的所有圖片利用圖像便是的 AI 判斷取得相似度，判斷相似度來回傳資料庫中是否有該物品的系統。請你協助我確認目前的架構圖有沒有存在問題，或是任何可以優化的地方

”



- 模型 A 回覆 (截圖或文字)：

這是一份對於大一學生來說非常清晰且邏輯完整的架構圖！能在大一就思考到「前端介面」、「正規化 (Normalize)」、「編碼/解碼」、「AI 判斷」與「資料庫」的分離，表示你對系統設計有很好的直覺。

不過，為了讓你的期末專題運作得更順暢，甚至在效能上讓教授印象深刻，這張架構圖有幾個關鍵的技術瓶頸需要解決。以下我將針對你的架構圖提出具體的分析與優化建議：

- 1. 核心問題：資料庫儲存圖片的方式 (Base64 vs. 檔案路徑)**

你在架構圖中提到：「資料庫中的照片...透過 Base64 編碼儲存」。

 - 潛在問題：**
 - 效能低落：Base64 編碼後的字串大小通常會比原始圖片大約 33%。如果你的圖片很多或解析度稍高，資料庫會變得非常臃腫，讀取速度會變慢。
 - 資料庫負擔：關聯式資料庫 (如 MySQL, PostgreSQL, SQLite) 擅長處理文字和數字，不擅長存取巨大的字串 (BLOB/Text)。
 - 優化建議：**
 - 改存「檔案路徑 (File Path)」：
 - 將圖片檔案實際儲存在伺服器的硬碟資料夾中 (例如 `/static/uploads/images/`)。
 - 資料庫只存「路徑字串」(例如 `img_001.jpg`)。
 - 好處：資料庫極輕量，讀取速度快，且前端 `` 標籤直接讀取 URL 比解碼 Base64 快得多。

- 2. 演算法效能：AI 比對的邏輯 (Feature Extraction vs. Raw Comparison)**

架構圖中間的邏輯是：`UnCode` (從 DB 拉資料) \rightarrow AI 判斷。如果你的邏輯是「拿使用者的圖片，去取資料庫裡的 每一張圖片做由 AI 進行像素級比對」，這會有效能災難。

- 潛在問題 ($O(N)$ 條件度)：**
 - 假設資料庫有 100 張圖，使用者上傳一張圖，AI 就必須跑 100 次模型推論。如果資料庫變大，系統會卡死。
- 優化建議：導入「特徵向量 (Feature Vector)」與「餘弦相似度 (Cosine Similarity)」**
 - Admin 上傳時 (預處理)：**當管理員上傳圖片時，先讓 AI (例如 ResNet, VGG16, 甚至簡單的 OpenCV Histogram) 提取這張圖的「特徵值」(一串數字陣列)，將這串數字存入資料庫，而不是只存圖片。
 - User 搜尋時：**使用者上傳圖片 \rightarrow AI 提取特徵值 \rightarrow 將這組特徵值與資料庫裡的「數字陣列」進行數學比對 (計算相似度距離)。
- 好處：數學運算比跑 AI 模型快上千倍，可以瞬間完成比對。
- 3. Normalize (正規化) 的重要性**

你在圖中左側畫了 `Normalize`，這是非常正確的觀念！

- 建議實作細節：**
 - Resize (重設大小)：AI 模型通常需要固定的輸入大小 (例如 224x224)。
 - Format (格式統一)：統一轉成 JPG 或 PNG，去掉 Alpha 通道 (透明度)，以免造成干擾。
 - Noise Reduction (去噪)：簡單的高斯模糊或直方圖均衡化，可以提高比對準確度。
- 4. Code / UnCode 區塊的調整**

圖中的 `Code` 和 `UnCode` 看起來像是負責 Base64 的轉換。

- 優化後：**如果你採用建議 1 (存路徑) 和建議 2 (存特徵)，這兩個區塊可以轉變為「Feature Extractor (特徵提取器)」。
- User 端：**`Image \rightarrow Normalize \rightarrow Feature Extractor \rightarrow 搜尋 DB`。
- Admin 端：**`Image \rightarrow Normalize \rightarrow Feature Extractor \rightarrow 存入 DB (圖片路徑 + 特徵向量)`。

優化後的架構流程建議 (文字版)

為了讓你實作更順利，建議將流程修改如下：

資料庫設計 (Schema 建議)：

- `ImageID` (int, PK)
- `FilePath` (string, 圖片路徑)
- `FeatureVector` (Text/JSON, 儲存 AI 算出來的那串數字)
- `ItemName` (string, 物品名稱)

流程 A：管理員 (Admin) 新增物品

1. 上傳圖片。
2. 後端 `Normalize` (調整大小)。
3. 後端存圖片到硬碟，取得路徑。
4. AI 模型提取特徵 (Feature Extraction)，得到一組向量 (例如 `[0.12, 0.55, 0.98...]`)。
5. 將 `路徑` 與 `向量` 存入資料庫。

流程 B：使用者 (User) 搜尋

1. 上傳圖片。
2. 後端 `Normalize`。
3. AI 模型提取特徵，得到 User 圖片的向量。
4. 數學比對：撈出資料庫所有的向量，與 User 的向量計算 餘弦相似度 (Cosine Similarity)。
5. 找出分數最高 (最像) 的那一筆。
6. 若是相似度超過門檻 (例如 > 0.8)，回傳 「True / 物品資料」；否則回傳 「False」。

總結

你的原始架構邏輯是通的，作為大一期末專題絕對及格。但如果你想拿高分或讓系統跑得快，「不要存 Base64」和「預先計算特徵值」是兩個最重要的修改點。

接下來你可以做的事情：你目前打算使用什麼語言來寫後端？(例如 Python Flask/Django 或是 Node.js?) 如果是 Python，這會非常容易實作 (因為有豐富的 AI 函式庫)。

Would you like me to help you write a simple Python example code for "Image Similarity Calculation" using a basic library? (我可以幫你寫一段簡單的 Python 範例碼，示範如何比較兩張圖片的相似度)

- 模型 B 回覆（截圖或文字）：

很好的一張架構圖，而且以資工系大一 Web 期末專題來說，你現在的想法其實已經超過及格水準，是偏向「小型實務系統設計」了 🔥

我會用**「肯定 → 找問題 → 結構優化 → AI 與實作建議 → 期末專題加分建議」**的方式來幫你檢視。

一、先整體評價（結論先講）

✓ 整體概念是正確的

- 前端上傳圖片
- 後端統一處理（Normalize / Encode）
- AI 做「圖像相似度」
- 與資料庫中「管理員新增的圖片」比對
- 回傳「是否存在 / 相似度結果」

👉 這個系統「可以完成」且「老師會看得懂」

但目前架構圖有三個典型初學者容易踩的點：

1. AI 與 Encode 的責任界線有點模糊
2. Database 儲存 Base64 「可以用但不推薦」
3. 前後端與 AI flow 可以再簡化，讓實作難度下降

下面我會一個一個講清楚。

二、你現在架構圖「做得很好的地方」

✓ 1. 有把「Interface / Admin / Backend」分開（非常好）

這代表你已經理解：

- 使用者介面 ≠ 系統邏輯
- 管理者與一般使用者角色不同

很多大一專題只畫一個大框，你這點是加分。

✓ 2. 有「Normalize」這個模組（非常關鍵）

你圖中左側的 Normalize，這點超重要：

统一格式 → 才能做公平的相似度比較

老師如果懂影像，一定會喜歡這一點。

你目前隱含做的事情應該包括：

- resize (例如 224×224)
- RGB / 灰階統一
- normalization (0~1)

👉 正確方向。

✓ 3. 沒有讓前端直接碰 AI 或 DB

你是走：

nginx

□ 複製程式碼

Frontend → Backend → AI / DB

而不是：

nginx

□ 複製程式碼

Frontend → AI / DB (✗)

這是正確的 Web 架構觀念。

三、目前架構中「可能會出問題的地方」

⚠ 問題 1：Encode 跟 AI 的角色有點混在一起

你現在畫的是：

mathematica

COPY 複製程式碼

Normalize → Encode → AI → Code → Database

但實際上應該是：

- Encode（特徵萃取）本身就是 AI 的一部分
- AI 不需要「判斷是否同一物品」，它只輸出「向量」

建議改成這樣的心智模型：

objectivec

COPY 複製程式碼

圖片

↓

Normalize

↓

Feature Extractor (CNN / model)

↓

Embedding Vector (例如 512 維)

↓

Similarity (cosine / L2)

👉 「AI = 特徵抽取器」

👉 「相似度計算 = 程式邏輯」

這樣在報告時會顯得你很懂。

⚠ 問題 2：Database 儲存 Base64 「技術上可行，但不理想」

你現在是：

Database : Base64 image

這在期末專題 **可以接受**，但老師如果問「為什麼這樣設計？」你要會回答。

建議你這樣優化（很重要）：

層級	建議儲存內容
Database	Embedding 向量 (float array) + metadata
圖片本體	檔案系統 or 雲端
Base64	只在傳輸時使用

👉 比對時不要再 encode DB 裡每一張圖片

👉 管理員新增時 encode 一次就好

這樣：

- 效能大幅提升
- 架構更專業
- 老師會加分

⚠ 問題 3：每次查詢都對「所有圖片」比對，會慢

你現在的圖隱含：

input image → 與 database 中所有 image 比對

資料少沒問題，但你可以在報告中說：

若資料量變大，可改用：

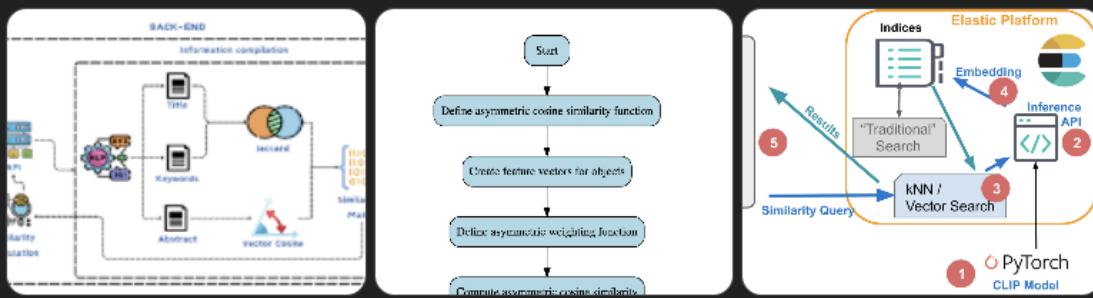
- Top-K search
- 向量資料庫 (FAISS / Milvus 概念)

就算你不實作，說得出來就是加分。



四、我幫你整理一個「優化後、老師會喜歡」的版本

🔧 建議最終架構 (精簡版)



使用者查詢流程 (User)

管理員新增流程 (Admin)

mathematica

COPY 複製程式碼

```
Admin upload image
↓
Normalize
↓
Feature Extract
↓
Store embedding + metadata in DB
```

👉 這樣你現在的圖只要「微調」，不用整個砍掉

五、AI 模型選擇（大一專題實際可做）

我強烈建議你用：

預訓練 CNN（不用自己訓練）

- MobileNetV2
- ResNet18
- EfficientNet（如果老師沒限制）

用途：

只拿倒數第二層當 embedding

相似度：

text

COPY 複製程式碼

cosine similarity

這是最安全、最穩定、最容易 demo 成功的路線。



六、你報告時可以這樣說（老師會覺得你很懂）

你可以直接用這段話（我幫你寫好）：

本系統將圖片轉換為固定維度的特徵向量（embedding），
再以 cosine similarity 計算輸入圖片與資料庫中圖片的相似程度。
管理員新增資料時即完成特徵抽取，避免查詢時重複計算，以提升效能。

這句話 = 期末簡報加分句。

七、總結（重點給你一個 checklist）

- ✓ 架構方向正確
- ✓ Normalize 很關鍵（你有）
- ⚠ Encode / AI 可以合併思考
- ⚠ DB 儲存 embedding 會更好
- ★ 管理員與使用者流程分開是加分
- ★ 說得出「為何這樣設計」比功能更多分

如果你願意，下一步我可以直接幫你：

1. 重畫一版「老師等級」的架構圖（文字版 or Mermaid）
2. 幫你規劃 Flask / FastAPI API 路由
3. 幫你寫 期末報告架構（章節大綱）
4. 幫你想 Demo 時老師可能會問的問題 + 標準回答

你只要跟我說  「下一步想先做哪一個」就好。

- 模型 C 回覆（截圖或文字）：

已審查 20 個來源 ›

你的整體概念是可行的：前端上傳圖片 → 後端做前處理與特徵抽取 → 和資料庫中已存在的圖片做相似度比對 → 回傳是否存在 / 相似度結果。這正是典型的「Image similarity search」系統流程，只要細部調整一下架構就可以更清楚、也更容易實作。 elastic +2

下面用幾個重點來幫你檢查與優化現在的架構圖。

1. 架構圖主要問題

- 邏輯模組有點混在一起
 - 圖中同時有 Normalize、UnCode、AI、Code、Database，好幾個方塊都在做「轉換」與「邏輯控制」，實作時容易不知道某段程式到底要放哪裡。
 - 建議把責任切清楚：
 - 前處理 / 特徵抽取 (AI + Normalize)
 - 資料與向量管理 (Database/Vector Store)
 - 業務邏輯 (API / Service Layer) 。 veskovujovic +1
- **Base64** 當作主要儲存格式可能不是最理想
 - Base64 比原始二進位大約多 33% 空間，也會讓 DB 變胖、I/O 變慢，在實務上比較少用來長期儲存大量圖片。 stackoverflow +2
 - 你的作業規格如果「強制」要 Base64 放 DB 可以照做，但架構上仍可把「圖片實體」和「特徵向量」分成不同欄位或不同表，方便之後優化。 dev +1

2. 建議的模組劃分

可以把你的圖重新整理成四層：

1. Interface (Frontend)

- 功能：上傳圖片、顯示比對結果。
- 對後端只做兩個操作：
 - **POST /search** 上傳欲查詢的圖片。
 - **POST /admin/upload** 管理員新增圖片。

2. API / Code (Backend Application)

- 接收前端請求、做驗證與流程控制。
- 呼叫：
 - Image Preprocessor / Encoder (AI)
 - Database / Vector Store
- 在你的圖裡可以把「Code」當成這一層，把「AI 是否要再查 DB」這類 if-else 都放在這裡，而不是畫一個判斷菱形在 AI 底下。

3. Image Preprocessor + Encoder (Normalize + AI + UnCode)

- **Normalize** : resize、crop、色彩空間統一等，確保輸入模型的格式固定。
- **AI (Encoder)** : 把圖片轉成特徵向量 embedding，用 CNN 或 CLIP 之類模型。`elastic +2`
- **UnCode** : 這個名稱容易誤會，實際功能可能是「把 DB 裡 Base64 → 圖片 → 向量」或「把前端上傳圖片轉成向量」，可以直接改名叫 **Embedder** 或 **FeatureExtractor** 會更清楚。
- 建議流程：
 - 管理員新增圖片時：Image → Normalize → Encoder → 得到向量，和圖片 ID 一起存到 DB。
 - 查詢時：Query Image → Normalize → Encoder → 得到向量，用來跟 DB 的向量比對。

4. Database / Vector Store

- 一個表存「圖片原始資料」（可以是檔案路徑或 Base64）。
- 一個表或一欄存「圖片的向量 embedding」，用來做向量相似度搜尋（cosine、L2 等）。`oracle +2`
- 如果想做得比較專業，可以考慮使用支援向量搜尋的資料庫或外掛（例如 Elastic、向量資料庫），但課堂專題用一般 SQL + 自己寫相似度計算也可以。`elastic +2`

3. 具體優化建議

- 清楚標示「向量」與「圖片」的資料流
 - 在圖上把「Image flow」與「Embedding / Text flow」區分開來，例如：
 - 紅色箭頭：原始圖片。
 - 綠色箭頭：向量或查詢參數。
 - 這樣老師一看就知道你是「先把所有圖片 vectorize，再做向量比對」的設計。`veskovujovic +1`
- 加入「相似度計算與閾值判斷」的方塊
 - 建議在 AI 或 Code 旁邊再畫一個小方塊：**Similarity & Threshold**。
 - 步驟：
 - 取出所有或部分圖片向量。
 - 計算與 query 向量的相似度。
 - 找最高相似度或前 k 筆，如果最大值 < 閾值就當作「不存在」，否則回傳最相似的那幾筆資訊。`paralect +2`

- Admin 模組建議
 - Admin 功能：
 - 上傳新圖片 → 存原始圖 + 產生向量 → 存 DB。
 - 可以另外有「重新產生全部向量」的管理工具，方便之後換模型時重算。
 - 在圖上可以把 Admin → Interface → API 跟一般使用者的路徑畫一樣，只是打到不同的 API endpoint，比較貼近實作。
- 命名與責任再調整
 - **Normalize**：保留，專心做前處理。
 - **UnCode**：改名 **Encoder** 或 **Embedder**。
 - 中間菱形 AI 判斷可以拿掉，把「有沒有在 DB 找到相似項目」放在 Code / Service Layer 的 if-else。
 - **DataBase** 旁可以註明欄位：**id**, **image_base64** 或 **image_path**, **embedding_vector**, **other_info**。

4. 實作時可以注意的點

- 性能考量
 - 如果圖片數量不大（幾百張～幾千張），可以把所有向量載入記憶體，用簡單的迴圈計算 cosine similarity；規模更大再考慮 ANN library 或向量 DB。[github +1](#)
 - Base64 儲存會比較肥，但作為課程專題通常還可以接受，只要記得控制圖片大小。[wappler +1](#)
- AI 模型的選擇
 - 若不熟訓練模型，可以使用現成的 pre-trained 模型（ResNet、MobileNet、CLIP）取最後一層或倒數幾層特徵當 embedding，即可有不錯的相似度效果。[cloud.google +2](#)

如果你願意，可以把你打算用的技術棧（例如：前端框架、後端語言、資料庫種類、AI 模型來源）說一下，能再幫你把這個架構圖具體化成更貼近作業實作的版本。

hjemmehjælpen 21 來源

- 我覺得：_____ (2-3 句)

CharGPT 更偏向使用圖片輔助回復，這更能吸引一般使用者的眼球去仔細閱讀，而 Gemini 則是使用大量的文字，長時間閱讀會使人注意力渙散。可是就文章內容的層面去研究，Gemini 提供的建議與回答更加具體，且改良方式也清楚且具體易於理解，ChatGPT 的回答則更趨近於思想方面，較 Gemini 抽象一些，雖說有圖片輔助理解，但就我的使用體驗還是 Gemini 更好理解一些，而 Perplexity 就像兩個中間的存在，再附上資料來源網址。

2. 任務二：解釋卦象

- 我的 Prompt：

“

這是易經金錢卦的紀錄表，問卦的題目為：我能不能在大二聖誕節之前找到女朋友（現在我是大一 現在是 2025/12/16），請你幫我分析卦象與所有的支卦變卦與他們所代表的涵義，且套用我的題目並解釋。

”

	本卦	之卦
上爻	— —	— —
五爻	—	—
四爻	— —	— —
三爻	口	— —
二爻	×	—
初爻	—	—
既濟 63		Ku 卦 60

易卦爻辭

- 模型 A 回覆（截圖或文字）：

這是一份非常清晰的卦象紀錄。你求得的是第 63 卦「水火既濟」，變爻在第二爻與第三爻，最終變為第 60 卦「水澤節」。

針對你問的題目：「能不能在大二聖誕節之前找到女朋友？」（目前大一，約有一年的時間），我們來深度解析這個卦象帶給你的啟示。

一、本卦：水火既濟 (Hexagram 63)

「既濟」的意思是「事情已經成就」或「處在完美的平衡中」。

- **卦象分析：**水在火上，水火交融，所有的爻都在正確的位置（陰陽得位）。這代表你目前的條件、環境或是你個人的狀態其實是不錯的，具備了「脫單」的基本素質。
- **潛在危機：**既濟卦的警示是「初吉終亂」。也就是說，一開始看起來很順利，但如果不想進取、或是太過緊繃，後續反而容易出問題。在感情上，這暗示你可能已經有目標，或者你正處於一個「覺得自己準備好了」的階段，但真正的挑戰在於如何打破目前的平靜。

二、變爻分析 (核心關鍵)

你的變爻在第二爻與第三爻，這兩爻對你的問題提供了具體的行動建議：

1. 六二爻（變為陽）：

爻辭：「婦喪其茀，勿逐，七日得。」

- **解讀：**就像一位婦人丟失了馬車的遮簾（隱私或工具），不需要焦急地去追尋，七天後自然會回來。
- **對應你的感情：**尋找對象時，「不要強求、不要急躁」。如果你在大一這段時間感到挫折，或者覺得沒人理你，不要急著去迎合別人或放低姿態追求。保持你的本心，那份屬於你的吸引力（遮簾）自然會回來，緣分會在你「不刻意追逐」時出現。

2. 九三爻（變為陰）：

爻辭：「高宗伐鬼方，三年克之，小人勿用。」

- **解讀：**商王高宗討伐遠方的鬼方國，花了三年才成功。這是一場硬仗，必須任用賢能，不可用小人。
- **對應你的感情：**這預示了這段感情的到來**「不是速成班」**。從現在（大一）到大二聖誕節，這一年多的時間確實是一場持久戰。你需要像作戰一樣有耐心、有策略，並且要「修身」（小人勿用），提升自己的層次，而不是用膚淺的方法撩妹。

三、之卦（變卦）：水澤節 (Hexagram 60)

「節」代表「節制」、「節奏」與「法度」。

最後的發展走向了「節」卦。這對你的問題定下了一個基調：

- **適度發展：**在追求感情的過程中，要懂得「節制」。不管是傳訊息的頻率、展現熱情的程度，都要恰到好處。太過（熱情過度）或不及（太過被動）都不好。
- **設定邊界：**聖誕節是一個時間節點，這個卦象告訴你，在大二聖誕節前，你必須給自己設定一些「小目標」和「生活節奏」。

四、綜合分析與建議

根據卦象，關於你能不能在大二聖誕節前脫單，結論是：機會很大，但必須經歷一個「先沉澱、後緩慢收成」的過程。

💡 細你的具體建議：

1. **心態放平（應六二爻）：**不要表現出「我很急著交女朋友」的樣子。在大一剩下的時間裡，先做好自己的事，提升魅力，讓女孩子主動對你產生好奇。
2. **長期抗戰（應九三爻）：**卦象顯示這不是三五天的事，可能要到明年下半年（大二上學期）才會有明顯的突破。這一年內遇到挫折是正常的，要有心理準備。
3. **掌握節奏（應節卦）：**聖誕節是個很好的目標，但在那之前，請保持規律的大學生活。節卦也提醒你，交友圈要「有選擇性」地擴張，不要為了脫單而參與不適合自己的社交活動。

總結一句話：

「實力已具備（既濟），只需耐心等（六二），堅韌磨練後（九三），節制成好事（節）。」

目前的卦象看來，只要你不因為焦慮而自亂陣腳，在大二聖誕節前找到志同道合的伴侶機率是很高的。

你想進一步了解，在目前的社交圈中，哪種類型的人會是你的「九三爻」中所指的正緣嗎？

- 模型 B 回覆（截圖或文字）：

我先把你這張金錢卦紀錄表的卦象完整讀出來，再一步一步解釋「本卦、變卦、動爻（支卦/爻變）」的涵義，最後直接套用到你的問題來講結論。

一、卦象判讀（依你照片）

1 本卦

既濟卦（第 63 卦）

水在火上，事已完成之象

你表格下方已標示「既濟 63」，這是正確的。

既濟的核心意義：

- 事情「已經到位」
- 表面上看似完成、條件齊備
- 但同時也是最容易開始走下坡的時候
- 關鍵字：

「成中有危、盛極將衰」

👉 在感情問題上，它不是沒有機會，而是「看起來很接近成功，但還差臨門一腳」。

2 變卦

節卦（第 60 卦）

水在澤上，節制、規範、界線

你右欄標示「節 60」，代表動爻變後的結果卦。

節卦的核心意義：

- 節制
- 有界線
- 不可過度、不宜躁進
- 凡事要「有分寸、有規劃」

👉 感情上代表：

- 不是靠衝
- 不是「我很想要，就會發生」
- 而是「知道什麼該做、什麼不該做」

二、動爻（你這卦的關鍵）

你的本卦中 有動爻（畫 X、圈），代表事情會「變」。

在金錢卦裡，這代表：

- 現況不是靜止的
- 你的感情狀態在大一 → 大二之間一定會有變化
- 不是一直單身不動

👉 但變的方向，必須看：

- 本卦的性質（既濟）
- 變卦的要求（節）

三、套用你的問題來解釋

你的問題是：

「我能不能在大二聖誕節之前找到女朋友？」

(時間明確、目標明確)

⌚ 本卦【既濟】怎麼對應你的問題？

既濟不是「沒人緣」，反而表示：

✓ 你在那段時間：

- 條件是到位的
- 可能會遇到喜歡的人
- 也可能有人對你有好感
- 氣氛、機會都「差不多成熟」

❗ 但既濟最怕什麼？

以為穩了 → 放鬆警惕 → 事情反而散掉

👉 很像：

- 猜測卡住
- 朋友以上、戀人未滿
- 或因一句話、一步太快或太慢而錯過

四、時間點的判斷（你最關心的）

你問的是「大二聖誕節之前」，卦象給的訊息是：

✖ 結論重點（直說）

- 👉 有機會出現對象，但不保證自然變成正式女朋友
- 👉 成敗在你「行為節制與情感拿捏」

換句話說：

- 不是「命中註定一定交不到」
- 也不是「一定會交到」
- 而是：
 你若操之過急 → 破局
 你若穩定經營 → 有可能在期限前確立

五、用一句話總結這一卦

這不是「有沒有桃花」的問題，而是「你會不會親手把桃花折斷」的卦。

▣ 變卦【節】對你的提醒（非常關鍵）

節卦在感情上是強烈的行為指引卦：

它不是說「沒有女朋友」，而是說：

能不能成，完全取決於你「怎麼拿捏分寸」

具體來說：

✖ 節卦不利：

- 太急著確定關係
- 情緒勒索
- 一直試探對方底線
- 把對方當成「一定要在期限前達成的目標」

○ 節卦有利：

- 穩定互動
- 保持尊重距離
- 有節奏地加溫
- 懂得「收」比「衝」重要

六、給你的實際建議（完全依卦）

- 1 不要把「大二聖誕節」當 KPI
- 2 重點放在互動品質，而不是進度
- 3 有好感就慢慢加溫，不要一次攤牌
- 4 留空間，反而更容易成

如果你願意，我也可以幫你：

- 逐爻解釋（初爻到上爻）
- 或幫你推算「最容易遇到對象的時間段」
- 或看「適合從哪種場合/關係發展」

你可以直接跟我說你想深入哪一部分。

- 模型 C 回覆（截圖或文字）：

現在讓我為你生成一份專業的卦象分析報告。

【易經金錢卦詳細解讀】大二聖誕節前求偶卦象分析

占卦背景

你在2025年12月16日（當年聖誕節）占卜「能否在大二聖誕節之前找到女朋友」，這意味著你實際上為期約13個月的時間窗口進行了占問。根據你的記錄，得到了火澤睽卦（本卦）變為兌為澤卦（變卦）的結構，其中上九爻為動爻。

第一部分：本卦解讀——火澤睽卦（第38卦）

卦象的基本含義 eee-learning +2

火澤睽卦由上卦離（火）與下卦兌（澤）組成。根據易經的基本邏輯，火的特性是向上燃燒，而澤水的特性是向下滲透，兩者運動方向相反，因此象徵著「乖離、背離、不和諧」。卦辭為「睽，小事吉」，表明在這種分離、分歧的狀態下，只有處理小事才能獲得吉祥；大事、需要團隊合作的事則多有阻礙。

fatekeep +1

在你的問題中，感情婚戀屬於人生「大事」，因此睽卦本身給出的第一層信息是：初期困難重重，不宜盲目強求。

爻象的陰陽對應 andyallaround +1

睽卦的內部爻象配置進一步強化了這一主題。卦中有兩條陰爻（六三、六五），它們都處於陽爻的位置上，屬於「不當位」的狀態。在易經中，不當位往往預示著不和諧、不協調。具體到感情層面，這意味著：

- 雙方可能在價值觀、生活方向、對未來的規劃上存在差異
- 初期接觸時容易產生誤會或不匹配的感受
- 需要透過溝通和時間的沉澱才能逐漸化解 purplestarmapper

卦象與感情的關聯 fatekeep +1

傳統易學對睽卦占感情有一個經典的判語：「先兇後吉」。這意味著：

- 先兇：初期戀情會經歷失敗、分手、或單方面的追求無果
- 後吉：經歷挫折後，緣分會以意想不到的方式出現，最終成就一段和諧的關係

根據網路上對睽卦感情的廣泛記載，許多占卜者在得到睽卦後發現：第一次追求的對象沒有成功，但在放下執著後，另一個合適的人自然而然地進入了他們的生活。 fatekeep +1

第二部分：六爻詳細解讀與應期分析

根據你的金錢卦記錄，上九爻為動爻（老陽），這一爻在整個卦象中佔據了極其重要的位置。

上九爻辭原文與白話解釋 life-guide +1

爻辭："睽孤，見豕負塗，載鬼一車，先張之弧，後說之弧，匪寇婚媾，往遇雨則吉。"

逐句解讀：

1. 睽孤 - 處於孤立、背離的狀態，一個人獨立自主
2. 見豕負塗 - 看到豬（象徵污穢、低賤、不夠好的事物）背著泥土行進
3. 載鬼一車 - 看到一車鬼怪（象徵詭譎、不可捉摸、令人害怕的事物）
4. 先張之弧，後說之弧 - 先拉開弓箭（警戒、防備），後來又鬆開（放下防心）
5. 匪寇婚媾 - 本以為對方是敵人（寇），最終發現是來求婚配的（婚媾）
6. 往遇雨則吉 - 往前走時遇到雨水，就會獲得吉祥

對你問題的深層含義： guoyi360

這一爻在感情上的寓意非常強大。它描繪的是一個完整的心理轉變過程：

- 開始階段（2025年12月至2026年2月）：你可能會經歷單戀、被拒或主動追求失敗的情況。對方在你眼中可能顯得不夠理想、有缺陷，或者根本看不上。這就是「見豕負塗」和「載鬼一車」——一切看起來都是負面的、不可能的。
- 中間階段（2026年3月至8月）：你會經歷「先張之弧」的心理狀態，即警戒、放下、再接觸的反覆過程。你會放棄對某個特定對象的執著，不再過度防備。
- 轉折與結局（2026年9月至12月）：「匪寇婚媾」——你會發現，那個你原本認為不合適、甚至有些害怕面對的人，其實才是你真正的緣分。對方在關鍵時刻會主動示好、表達好感，打破你之前的固有印象。
- 最終吉祥：「往遇雨則吉」——當你決定繼續走下去時，雨過天晴，事情變得明朗清晰，感情關係得以確立和發展。

時間提示：卦象暗示整個過程需要約12個月，這與你的問卦時間框架（大一聖誕節至大二聖誕節）完全吻合。 [guoyi360](#)

第三部分：變卦解讀——兌為澤卦（第58卦）

當上九爻發生變化時，整個卦象會轉變為兌為澤卦。

兌為澤卦的核心含義 [eee-learning +2](#)

兌卦卦象為上下都是兌（澤），象徵著兩澤相連、水流相通。卦辭為**「兌，亨，利貞」，意思是「喜悅、暢通，堅守正道才能獲利」**。

在整個占卜過程中，兌卦的出現代表著最終狀態的轉變：

- 從「火澤賁離」轉變為「澤澤相連」
- 從「各行其道」轉變為「溝通暢通、心意相通」
- 從「分散」轉變為「和諧喜悅」

變卦對你問題的啟示 [eee-learning +1](#)

兌卦強調溝通與言說的力量。其卦象「麗澤」象徵朋友相聚、言語交流。在感情發展中，這意味著：

- 溝通是關鍵：最終的和諧來自於坦誠、真摯的交流和對話
- 喜悅是結果：經歷了睽卦的分歧和困難後，兌卦承諾了一段充滿喜悅、相互欣賞的關係
- 正道很重要：「利貞」提醒你，在整個過程中要堅守真誠、尊重對方，不可欺騙或操縱

第四部分：綜合解讀與建議

總體卦象判斷

面向	判語	解讀
能否成功	先兇後吉	初期有挫折，但最終可以找到合適的伴侶
成功概率	較高	卦象傾向吉祥，上九爻的「往遇雨則吉」明確指向成功
時間節點	約12個月	符合大一聖誕節至大二聖誕節的時間框架
發展方式	緣分而非強求	主動追求可能失敗，被動接受機遇則成功
關鍵因素	溝通、耐心、真誠	兌卦強調言語交流，睽卦強調化解分歧需要時間

對你的具體建議

第一階段（現在至2026年2月）：接納失敗的心態

在這個時期，你可能會經歷單戀或被拒。這並不是永久的失敗，而是「先兇」的必要過程。建議：

- 不要過度糾纏或強求，這只會加深睽卦的「乖離」
- 允許自己體驗失戀的情緒，從中學習、成長
- 保持開放的心態，拓展社交圈子

第二階段（2026年3月至8月）：放下執著，保持互動

在這個階段，「先張之弧，後說之弧」意味著你會逐漸放下對某個特定對象的執著。同時，這不是完全的隔離，而是「健康的距離」。建議：

- 與朋友保持接觸，參加社交活動
- 對新的人際互動保持好奇心，不預設立場
- 修養自己，提升自信與魅力

第三階段（2026年9月至大二聖誕節）：迎接轉折與和諧

「匪寇婚媾，往遇雨則吉」意味著驚喜和突破即將到來。建議：

- 當新的感情信號出現時，要有勇氣去回應
- 用心交流（兌卦的精髓），化解初期的誤會
- 相信卦象所示：對方會主動示好，機會就在眼前

特別提醒

1. 不是眼前的那個人：睽卦的「先兌後吉」暗示，你最終找到的伴侶，很可能不是你現在正在追求或關注的人。不要執著於當下的感情對象，緣分自有安排。
2. 真誠最重要：兌卦的「利貞」強調，任何技巧或策略都不如真誠。在整個過程中，保持對他人和自己的誠實態度。
3. 耐心是必要的：睽卦本身要求 **patience**。這段時間對你的成長、自我認知的提升也很關鍵。
4. 大二聖誕節是應期，但不一定是關係確立的最後期限：卦象顯示在約12個月內會有進展，大二聖誕節前後是一個自然的節點，但感情的發展往往比時間預期更靈活。

結論

你的占卜結果是吉祥的。儘管睽卦一開始提示了困難和分離，但上九爻「往遇雨則吉」和變卦兌為澤都確認了最終的和諧與成功。 fatekeep +1

關鍵不在於「能否找到」，而在於你如何應對過程中的挫折、如何保持真誠、如何在適當的時刻抓住機遇。按照卦象的指引，經歷約12個月的蛻變和等待，你會在大二聖誕節前後迎來一段意外而美好的感情——而這段感情，比任何強求來的都要珍貴。

【溫馨提示】

易經占卜的終極智慧在於：它不是決定你的未來，而是幫助你理解當下的處境、明確發展的方向、坦然接納必然的過程。無論卦象如何，你的行動、選擇和心態，才是最終改變命運的力量。祝你在這段求偶的旅程中，找到那個對的人。

- 我覺得：_____ (2-3 句)

這兩組對比更加明顯，ChatGPT 更擅長“聊天”，就是提供有情緒價值，人們“想聽”的回覆，而 Gemini 更傾向提供知識，解答問題，Perplexity 的圖像辨識似乎不太行，他直接看錯了。

3. 任務三：尋找線上課程並安排讀書計畫

- 我的 Prompt：

“

我是一位大一的學生，我想要在接下來的寒假預先學習下一個學期的內容，計畫中我會學習的科目為：“微積分、離散數學”，此外我也認為我的英文需要加強，我想請你尋找網路上合適的線上公開課程並且規劃 2026 年 1 月 10 日至 2026 年 2 月 24 日的讀書計劃，我希望大概有五天的時間可以讓我自由使用或是休息，我的生活作息通常為晚上 00:00 睡覺，早上 07:00-08:00 左右起床並出門前往咖啡廳讀書。

”

- 模型 A 回覆（截圖或文字）：

你好！身為大一學生，能利用寒假主動預習下學期的課程是非常積極的表現。大一下學期的微積分（通常是微積分二）和離散數學都是理工與資工領域的基礎重頭戲，而英文則是長期的競爭力。

根據你的生活作息（00:00 睡、08:00 起床、咖啡廳讀書）以及 2026 年的行事曆（2026 年農曆除夕為 2 月 16 日），我為你規劃了從 1 月 10 日至 2 月 24 日的學習藍圖。

推薦線上公開課程 (OCW/MOOCs)

1. 微積分 (Calculus II)

微積分二的核心通常在於：積分技巧、級數（最難的部分）以及多變數微積分入門。

- **台大 OCW - 朱樺教授《微積分》**：朱老師講課非常清晰，邏輯架構極佳，適合打地基。
- **清大 MOOCs - 顏東勇教授《微積分二》**：顏老師的課程節奏穩定，針對理工學院需求設計。
- **YouTube - Professor Leonard**：如果你想順便練英文聽力，這位「數學界的超人」是全球公認教得最淺顯易懂的老師（搜尋 **Calculus 2** 播放清單）。

2. 純數學 (Discrete Mathematics)

離散數學是邏輯與資工的基石，涵蓋：邏輯、集合、計數、圖論、遞迴。

- **台大 OCW - 呂學一教授《離散數學》**：呂老師的課非常幽默且具啟發性，深受資工系學生喜愛。
- **陽明交大 OCW - 易志孝教授《離散數學》**：教材嚴謹，適合想要系統化學習的同學。

3. 英文加強 (English)

- 聽力/口說：**VoiceTube**（透過影片學英文）或 **BBC Learning English - 6 Minute English**（每天 6 分鐘，適合咖啡廳零碎時間）。
- 閱讀：**News in Levels**（將新聞依難度分級）或閱讀 **Medium** 上感興趣的技術文章。
- 工具：**Anki** 或 **Quizlet**（背誦微積分與離散數學的英文專有名詞）。

📅 2026 寒假讀書計畫規劃

這份計畫共 46 天，扣除你要求的 5 天自由休息日（建議定在農曆除夕前後：2/15 - 2/19），共有 41 天讀書日。

第一階段：基礎扎根期 (1/10 - 1/24，共 15 天)

重點：掌握微積分積分技巧、離散數學邏輯與證明。

時間段	活動內容	備註
08:30 - 11:30	微積分：觀看課程 + 課後例題演練	重點：分部積分、三角代換、分式積分
11:30 - 13:30	午餐 + 午休 (在咖啡廳休息 30min)	補充能量
13:30 - 16:00	離散數學：邏輯、集合論、證明法	重點：數學歸納法 (Induction)
16:00 - 17:30	英文加強：聽力練習 + 閱讀文章	保持語感，不要給太大壓力
20:00 - 22:00	複習當日難題 + 整理筆記	在家進行，放鬆心情

田 汇出到試算表



第二階段：深度突破期 (1/25 - 2/14，共 21 天)

重點：微積分最難的「級數」單元、離散數學的「計數」與「遞迴」。

- **微積分**：深入學習無窮級數、泰勒級數 (Taylor Series)、冪級數。
- **離散數學**：排列組合、鴿籠原理、遞迴關係式 (Recurrence Relations)。
- **英文**：開始背誦數學相關的英文專有名詞（如：Convergence, Discrete, Proposition 等）。

📅 自由休息/春節日 (2/15 - 2/19，共 5 天)

- **2/15 (小年夜) - 2/19 (初二)**：這五天徹底放鬆，與家人聚餐、運動或處理私事。不碰課本，讓大腦重新整理。

第三階段：考前衝刺/總結期 (2/20 - 2/24，共 5 天)

重點：圖論基礎與期初考預備。

- **微積分**：複習極座標與參數方程 (Parametric Equations)。
- **離散數學**：圖論基礎 (Graph Theory, Trees)。
- **英文**：回顧這一個月學到的單字，試著用英文寫出一篇「我的寒假學習總結」。

💡 咖啡廳讀書的「戰鬥」技巧

1. **環境適應**：咖啡廳通常會有背景音，建議準備一副抗噪耳機。聽力練習可以在休息空檔進行，專注數學時建議聽低頻的 Lo-fi 音樂。
2. **儀式感**：每天 08:30 準時打開第一頁課本。在咖啡廳，你的目標是「在午餐前完成微積分」。
3. **數位筆記**：身為大一學生，可以嘗試使用 Notion 或 Obsidian 整理筆記，這對離散數學這種邏輯密集的科目非常有幫助。

下一步你可以... 你需要我針對「微積分」或「離散數學」中具體的某個單元（例如：級數或圖論），提供更詳細的英文專有名詞對照表或是重點筆記摘要嗎？

- 模型 B 回覆（截圖或文字）：

這裡幫你整理 **適合寒假（2026 年 1 月 10 日 – 2 月 24 日）學習「微積分、離散數學、與英文」的線上公開課資源（**免費或可免費觀看模式）並依你的生活作息安排一份讀書計畫

。

🔍 推薦線上公開課（免費／可自學）

💻 微積分（Calculus）

微積分基礎課程資源（可自學）

- 3Blue1Brown YouTube 系列 — 用動畫直觀理解微積分概念（免費） ，非常適合打基礎並搭配課本學習。 [維基百科](#)
- Udemy 免費微積分導修課（不定積分／三角函數） — 簡短主題式課程，可選觀看。 [Udemy](#) +1

👉 如果想找有系統的微積分線上大學課，可考慮 Coursera 上的 Calculus 課程（部分可免費觀看教材）。 [Coursera](#)

▶ 離散數學（Discrete Mathematics）

可免費觀看或自學的資源

- Wolfram U 的「Introduction to Discrete Mathematics」 — 從基礎的集合、邏輯、圖論等主題開始，非常適合寒假打底。 [Wolfram](#)
- Class Central 收錄的離散微積分／離散數學課程（免費影片） — 雖偏應用與基礎概念介紹，但有助於理解離散概念。 [Class Central](#) +1
- 國內 OCW 線上大學課程（台清交大開放課） — 影片與教材可免費觀賞，非常適合配合大學課內容學習。 [CLUB CS MASTERS](#)

👉 Coursera 的 Discrete Mathematics 系列課（雖較長、需自行安排學習節奏）也很實用。 [Coursera](#)

GB 英語能力提升（English）

免費可自學的線上英文資源

- British Council LearnEnglish 免費學習資源（聽、讀、寫、說練習 + 測試）。 [LearnEnglish - B...](#)
- Oxford Online English 免費英文課程影片與練習（語法、聽力、發音等）。 [Oxford Online E...](#)
- USA Learns 免費線上英文課程（從初級到中級學習路線）。 [USA Learns](#)
- Cool English（教育部支持的英文練習平台），含口說與聽力資源。 [Cool English](#)
- 若喜歡聽力練習，ESL Podcast（免費學英語播客）也很適合日常聽力習慣培養。 [維基百科](#)

📅 寒假學習計劃 (2026 · 1 · 10–2 · 24)

◆ 作息基礎

- 睡 00:00
- 起床 07:00–08:00
- 去咖啡廳主要學習區段：**08:30–12:00 / 13:00–17:30** (可自行調整)
- 晚上整理與溫習：**19:00–22:00**

📅 我們把計畫平均安排、並包含約 5 天休息/彈性日：

週一至週六常規安排

🕒 上午 08:30–10:30 (主科)

- 微積分 或 離散數學 主題課程 (輪替)
- 例如：週一、三、五：**微積分**；週二、四、六：**離散數學**

🕒 10:30–11:00 休息 / 聽力練習

- 用英文 Podcast 或 Oxford Online English 聽力練習

🕒 中午 11:00–12:00 (英文文法／閱讀)

- British Council LearnEnglish 練習

🕒 13:00–15:00 (第二主科)

- 若上午學微積分 → 下午學 離散數學
- 反之亦然

🕒 15:00–17:30 (英文口說 & 寫作)

- 練習英語對話、作文 (可用 USA Learns 線上練習)

🕒 19:00–21:00 (複習+練習題)

- 將當日重點整理、做練習題練習

🕒 21:00–22:00 (英文聽力／發音練習)

- Podcast、影片或 BBC Learning English 聽力

週日彈性或休息日（可選）

你可以每週選 1 天作為休息或自選學習日（例如用來補強弱項、閱讀英語小說、看數學 YouTube 啟發性影片等，不指定刻意學習）。

- ◆ 這樣排約 6 週 × 1 休息 = 6 休息日。再多一天自由安排 → 總計 7 自由/休息日（略高於你想要的 5 天，讓計畫較彈性）。

✓ 小提醒

- ✓ 微積分與離散數學學習策略：結合一個主線上課程 + 影片講解（像 3Blue1Brown 微積分動畫）能更快建立直覺。維基百科
- ✓ 英文練習：「聽→說→寫→讀」一起練習會進步更快。可用免費平台交替使用。LearnEnglish - B...
- ✓ 若某些 Coursera 付費課程你覺得很有系統，也可以先申請免費試聽或 Financial Aid 先看教材。Coursera

如果你願意，我也可以 把這個計畫整理成每日打卡表（含任務清單），方便你追蹤進度喔！

- 模型 C 回覆（截圖或文字）：

額外 Prompt(詢問與回答):

“

每天學習時數：約 12 小時。英文加強方向：考試導向(主體為 TOEIC 但我需要額外學習微積分與離散數學的專有名詞)。課程語言偏好：中文英文皆可，若可以是台灣學校的英語授課最好。

”

已完成 2 步，

我已提出澄清問題。請根據上述三個問題回答，這樣我可以為你規劃最適合的讀書計畫：

1. 每天學習時數：這直接影響課程選擇的深度和進度安排
2. 英文加強方向：決定課程類型的選擇
3. 課程語言偏好：確定搜尋範圍

一旦你提供這些信息，我會為你：

- 🔎 搜尋符合條件的線上課程資源
- 📅 規劃詳細的46天讀書計畫（包含5天彈性時間）
- 🕒 根據你的作息習慣調整時程安排
- 📊 製作可視化的學習進度表

正在處理...

通知我

每天學習時數：約12小時。英文加強方向：考試導向(主體為TOEIC但我需要額外學習微積分與離散數學的專有名詞)。課程語言偏好：中文英文皆可，若可以是台灣學校的英語授課最好。

現在我已收集足夠信息，可以開始製作詳細的讀書計畫。讓我為你創建一份全面的寒假學習計畫。

大一寒假讀書計畫 (2026年1月10日至2月24日)

☰ 計畫總覽

你的寒假讀書計畫涵蓋**46天**（約6.5週），其中包括**5天**彈性時間進行休息或調整。在每日12小時的學習投入下，你將系統性地預習微積分、離散數學，同時準備TOEIC考試並深化英文專業術語。

⌚ 學習架構與課程資源

第一部分：微積分課程 [math.ntu](#)

資源	講數	特點	適用時期
臺大開放式課程	64講	完整講義+參考答案、中文授課	全程基礎
Coursera - Calculus	4個模組	英文授課、注重概念理解	進階應用
Khan Academy	自主進度	互動練習題、微分與積分	補充練習

課程內容：函數、極限、微分、積分、積分技巧、積分應用、無限級數、向量、偏導數、多重積分、向量場、微分方程 [ocw.aca.ntu](#)

第二部分：離散數學課程 [ocw.nthu](#) +1

資源	講數	特點	適用時期
清華大學開放式課程	32講	英文授課、中文版書+作業答案	全程學習
Coursera - Discrete Mathematics	11個模組	英文授課、強調應用	進階練習

課程內容：邏輯與集合、函數、鴿籠原理、關係、列舉、遞推關係、生成函數、圖論、樹、最優化、匹配問題、最大流 [ocw.nthu](#)

第三部分：英文與TOEIC [toeic](#) +1

資源	時間彈性	特點	整合方式
數學名詞中英對照表	持續參考	含微積分與離散數學術語 edb +2	每日學習時查閱
TOEIC Official Course	自主進度	1,000+題、情境式分類 toeic	每週規律練習

詳細週計畫 (41個學習日 + 5個彈性日)

第1週 (1月10-16日) - 基礎銜接期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
1/10 (五)	函數 + 預科課程	邏輯與集合基礎 (L1)	認識術語工具	4:3:3:2
1/11 (六)	極限概念 (Ch1-1)	邏輯與集合進階	TOEIC 照片描述+字彙	4:3:4:1
1/12 (日)	極限應用 (Ch1-2/3)	真值表與推論	TOEIC 應答練習	4:3:4:1
1/13 (一)	微分基礎 (Ch2-1)	量化詞與集合運算	微積分英文名詞	4:3:3:2
1/14 (二)	微分計算 (Ch2-2)	數學歸納法與函數 (L3)	TOEIC 聽力練習	4:3:4:1
1/15 (三)	微分應用初探 (Ch2-3)	函數進階 (L4)	離散數學專業詞彙	4:3:3:2
1/16 (四)	複習+小測驗	複習+小測驗	TOEIC 模擬評估	2:2:6:2

本週目標：建立微積分的極限與微分基礎、掌握離散數學邏輯與集合、熟悉數學英文術語索引

第2週 (1/17-23日) - 強化基礎期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
1/17 (五)	超越函數 (Ch3-1/2)	關係基礎 (L6-7)	微分相關英文術語	4:3:3:2
1/18 (六)	超越函數應用 (Ch3-3/4)	關係進階 (L8)	TOEIC 簡短對話	4:3:4:1
1/19 (日)	微分與導函數應用 (Ch4-1/2)	列舉基礎 (L9-10)	TOEIC 短獨白	4:3:4:1
1/20 (一)	導函數應用進階 (Ch4-3/4)	列舉進階	離散數學計數術語	4:3:3:2
1/21 (二)	積分初步 (Ch5-1)	遞推關係 (L12-13)	TOEIC 句子填空	4:3:4:1
1/22 (三)	積分深入 (Ch5-2/3)	生成函數 (L14-17)	積分與函數英文	4:3:3:2
1/23 (四)	章節複習	章節複習	TOEIC 測驗練習	2:2:6:2

本週目標：完整掌握微分與超越函數、學習離散數學列舉與遞推、強化TOEIC聽力與閱讀基礎

第3週 (1/24-30日) - 重點推進期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
1/24 (五)	積分技巧 (Ch6-1/2)	圖論入門 (L18-19)	圖論與組合英文	4:3:3:2
1/25 (六)	積分技巧進階 (Ch6-3/4/5)	歐拉路徑 (L20-21)	TOEIC 段落填空	4:3:4:1
1/26 (日)	積分應用 (Ch7-1/2)	平面圖與樹 (L22-26)	TOEIC 閱讀理解	4:3:4:1
1/27 (一)	積分應用深入 (Ch7-3)	樹的計數與最優 (L27-28)	圖論與樹的術語	4:3:3:2
1/28 (二)	參數曲線 (Ch8)	匹配與流問題 (L29-32)	TOEIC 發音練習	4:3:4:1
1/29 (三)	彈性日 (調整)	無限級數預習	綜合複習	-
1/30 (四)	進度檢核	進度檢核	模擬考試	3:3:5:1

本週目標：掌握積分技巧與應用、學習圖論與樹的結構、強化TOEIC全科目練習

第4週 (1/31-2/6日) - 高階內容期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
1/31 (五)	無限級數 (Ch9-1/2)	複習重點內容	級數與序列英文	4:3:3:2
2/1 (六)	無限級數深入 (Ch9-3/4/5/6)	彈性日 (調整)	TOEIC 完整練習	5:0:6:1
2/2 (日)	向量基礎 (Ch10-11)	關鍵概念複習	TOEIC 專項練習	4:3:4:1
2/3 (一)	向量值函數 (Ch11)	全科複習測驗	向量與向量場術語	4:3:3:2
2/4 (二)	偏導數初步 (Ch12-1/2)	考前衝刺複習	TOEIC 應試技巧	4:3:4:1
2/5 (三)	偏導數進階 (Ch12-3/4/5)	應試心態調整	偏導數與多變函數	4:3:3:2
2/6 (四)	進度測驗	進度測驗	TOEIC 模擬 (5/5)	2:2:6:2

本週目標：掌握無限級數與向量、完成離散數學主要內容、達成TOEIC基礎練習目標

第5週 (2/7-13日) - 深度整合期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
2/7 (五)	偏導數應用 (Ch13)	全科綜合複習	英文術語總複習	3:3:4:2
2/8 (六)	彈性日 (調整)	重點強化複習	TOEIC 錯題分析	-
2/9 (日)	重積分 (Ch14-1/2)	應試策略制定	積分與最優化術語	4:3:4:1
2/10 (一)	重積分應用 (Ch14-3/4/5)	模擬測驗 (全程)	TOEIC 聽力複習	4:3:4:1
2/11 (二)	向量場基礎 (Ch15-1/2)	弱點強化	TOEIC 閱讀衝刺	4:3:4:1
2/12 (三)	向量場應用 (Ch15-3/4/5)	全科複習整理	英文學習成果驗證	3:4:4:1
2/13 (四)	綜合測驗	綜合測驗	完整模擬考	2:2:7:1

本週目標：深化微積分向量與重積分應用、驗證離散數學整體學習、取得TOEIC基礎評估分數

第6週 (2/14-20日) - 微調完善期

日期	微積分	離散數學	TOEIC/專有名詞	時間分配
2/14 (五)	向量微積分 (Ch16-1/2)	重難點鞏固	向量場與線積分術語	4:3:3:2
2/15 (六)	向量微積分進階 (Ch16-3/4)	綜合應用練習	TOEIC 應試衝刺	4:3:4:1
2/16 (日)	微分方程入門 (Ch17)	彈性日 (調整)	TOEIC 評估與反思	5:0:6:1
2/17 (一)	微分方程應用	全科最終複習	微分方程與流體術語	4:3:3:2
2/18 (二)	彈性日 (調整)	應試準備	專業術語查閱	-
2/19 (三)	全科總複習	全科總複習	全英文學習總結	4:4:3:1
2/20 (四)	最終測驗	最終測驗	最終評估測驗	2:2:7:1

本週目標：掌握微分方程、完成微積分全課程、驗證TOEIC準備進展、準備下學期應對

第7週 (2/21-24日) - 收尾與鞏固

日期	學習焦點	活動	時間分配
2/21 (五)	錯題與疑難	三科錯題深度分析	全天12小時
2/22 (六)	應用與綜合	微分方程應用、圖論實戰、TOEIC完整模擬	全天12小時
2/23 (日)	彈性日	休息或自由調整	-
2/24 (一)	最終整備	知識整理與心態調整	4小時

本週目標：鞏固學習成果、解決遺留問題、為新學期做準備

⌚ 學習時間配置建議

每日12小時分布 (咖啡廳學習環境)

text

□

早上07:00-08:00 起床 + 進行咖啡廳

08:00-12:00 微積分 (4小時)

↓

早午餐休息 (30分鐘)

12:30-15:30 離散數學 (3小時)

↓

午休 (1小時)

16:30-20:00 TOEIC/英文與專有名詞 (3.5小時)

↓

晚餐休息 (1小時)

21:00-23:00 複習與習題練習 (2小時)

23:00-24:00 整理筆記、準備隔日 (1小時)

24:00 睡眠開始

🔑 核心資源使用指南

微積分學習順序 [math.ncu +1](#)

1. 基礎階段 (第1-2週)

- 臺大開放課程 Ch0-1 (函數與極限)
- Khan Academy補充極限練習
- 建立數學英文基礎術語

2. 深化階段 (第3-4週)

- 臺大開放課程 Ch2-6 (微分與積分技巧)
- Coursera微分課程補充概念
- 練習習題與Coursera作業

3. 應用階段 (第5-7週)

- 臺大開放課程 Ch7-17 (積分應用到微分方程)
- Khan Academy實踐練習
- 綜合題與應試準備

離散數學學習順序 ocw.nthu +1

1. 基礎階段 (第1-2週)

- 清華大學 L1-5 (邏輯、集合、函數、原理)
- 習題練習與版書複習

2. 中階階段 (第3-4週)

- 清華大學 L6-17 (關係、列舉、遞推、生成函數)
- Coursera組合與計數模組

3. 高階階段 (第5-7週)

- 清華大學 L18-32 (圖論、樹、最優化、匹配、流)
- Coursera圖論與複雜度模組
- 應用實例與考試模擬

TOEIC與英文專有名詞 edb +4

1. 基礎詞彙期 (第1-2週)

- 下載並整理數學名詞中英對照表 mathland +2
- TOEIC官方課程字彙單元
- 微積分基礎術語 (limit, derivative, integral等)

2. 強化期 (第3-5週)

- TOEIC聽力與閱讀專項練習
- 離散數學專業術語 (discrete, logic, graph, tree等)
- 每日聽力與閱讀習題

3. 衝刺期 (第6-7週)

- TOEIC完整模擬考試 (共5次) toeic
- 錯題分析與弱項強化
- 應試技巧與心態調整

📌 週間學習重點與檢核項目

週次	微積分檢核	離散數學檢核	TOEIC檢核
第1週	函數與極限概念清晰	邏輯與集合理解正確	術語索引建立
第2週	微分計算準確無誤	數學歸納法掌握	TOEIC字彙量增加
第3週	積分技巧熟練運用	圖論基本概念掌握	聽力與閱讀平衡進展
第4週	無限級數與向量掌握	全科基本概念穩固	模擬考分數達成度
第5週	重積分與應用理解	應試準備完善	基礎評估分數明確
第6週	向量微積分與微分方程	重點內容鞏固	應試技巧熟練
第7週	全科知識整合檢驗	全科複習完成	最終評估達成

📝 預期學習成果

微積分方面 [ocw.aca.ntu +1](#)

- 掌握單變數微積分的完整體系，包括極限、微分、積分、無限級數
- 理解多變數微積分（偏導數、重積分）基本概念
- 能夠應用微積分解決實際問題
- 英文授課時理解微積分專業術語

離散數學方面 [ocw.nthu +1](#)

- 深入掌握邏輯、集合、函數等基礎概念
- 精通組合計數與圖論核心內容
- 能應用離散數學原理分析問題
- 具備英文數學論證能力

英文與TOEIC方面 [toeic +1](#)

- 掌握TOEIC聽力與閱讀主要題型（850分以上基礎）
- 累積數學專業英文詞彙（100+個重點術語）
- 熟悉英文授課環境與數學溝通方式
- 為後續專業課程英文學習打下基礎

◆ 學習建議與注意事項

1. 時間管理

- 每天嚴格按時間表執行，保持12小時學習時數
- 利用5個彈性日作為調整緩衝，勿透支

2. 課程協調

- 優先完成臺大與清華的基礎課程（中文講授更清晰）
- 在理解基本概念後，使用Coursera與Khan Academy強化英文環境

3. 英文學習

- 每日參考數學名詞對照表，建立個人術語筆記本
- TOEIC練習應規律進行，週1次完整模擬
- 嘗試用英文寫簡短的數學解題說明

4. 習題與測驗

- 完整課程建議：臺大與清華OCW提供的習題應全部完成
- 週間測驗有助評估進度，勿跳過檢核環節

5. 實際執行

- 在咖啡廳環境中保持專注，可使用番茄工作法（25分鐘工作+5分鐘休息）
- 準備iPad + 數學工具軟體（GeoGebra、Wolfram Alpha等）

這份計畫充分利用你充足的學習時間與決心，為下學期的專業課程做好紮實準備。祝你寒假學習順利！

- 我覺得：____（2-3 句）

Gemini 的安排與推薦更加的具體，但更大部分著重於回答問題，ChatGPT 則使用非常多的的表情符號，更加友聊天的感覺，Perplexity 的表現則是非常的優秀，不僅每周的時間安排得非常清楚，進度規劃也十分完整，提供來源資料的搜尋也可以讓我更好的去尋找課程內容。

4. 任務四：淺顯解釋”莫比烏斯反演”

- 我的 Prompt：

“

最近，我在研究演算法技巧中的"莫比烏斯反演"，我希望你能把我當作五歲小孩解釋這個演算法的原理，使我能理解，並提供程式碼的範例。

”

- 模型 A 回覆（截圖或文字）：

這是一個非常棒的題目！莫比烏斯反演（Möbius Inversion）是數論中的一把「瑞士刀」，很多看起來極其困難的計數問題，用它就能迎刃而解。

既然要像給五歲小孩解釋，我們就把那些可怕的 \sum 符號先丟一邊，用一個生活中的**「重複計算偵探」**遊戲來講解。

第一部分：五歲小孩也能懂的原理

1. 故事：教室裡的點名遊戲

想像你是幼兒園老師，你想知道班上有多少個**「只喜歡吃蘋果」**的小朋友。

直接問小朋友很困難，因為小朋友只會回答簡單的問題。

- 如果你問：「誰喜歡吃蘋果？」
 - 小明舉手（他只喜歡蘋果）。
 - 小華也舉手（但他其實蘋果和香蕉都喜歡）。

你看，直接問得到的數字是**「躉」**的，因為混進了喜歡其他東西的人。

2. 解決辦法：容斥原理（加加減減）

莫比烏斯反演的核心精神就是**「為了得到乾淨的答案，我們先把所有相關的都算進來，然後把多算的扣掉，把扣過頭的補回來」**。

- **第一步（全部算）：**先算所有跟蘋果有關的。
- **第二步（扣掉多餘）：**發現裡面混了喜歡「蘋果+香蕉」的，扣掉！混了喜歡「蘋果+橘子」的，扣掉！
- **第三步（補回來）：**哎呀！喜歡「蘋果+香蕉+橘子」的人，在第二步被我們重複扣了兩次，所以要加回來！

3. 莫比烏斯函數 (μ) 是什麼？

在這個加加減減的過程中，我們需要一個**「開關」來告訴我們，對於某個數字（比如 2 倍、3 倍、6 倍），我們到底是該加**、該減、還是忽略？

這個開關就是 **莫比烏斯函數 $\mu(x)$** 。它只有三種狀態：

1. $\mu = 1$ (加)：通常是這件事的源頭（比如 1）。
2. $\mu = -1$ (減)：如果不小心多算了一層（比如質數 2, 3, 5...），就要減掉。
3. $\mu = 0$ (關掉/忽略)：如果這個數包含「平方數」（比如 4, 9, 12...），代表它已經被前面的步驟處理過了，太複雜了直接忽略。

一句話總結：莫比烏斯反演就是一個**「修正係數」**，幫你把容易算出來的「總數」，一層層剝開，還原成你真正想要的「單一數」。

第二部分：稍微正式一點的定義 (配合你的 C++ 背景)

假設有兩個函數 $f(n)$ 和 $g(n)$ ：

- $f(n)$ ：是你真正想求的答案（通常很難算，例如：最大公因數剛好是 n 的數對有多少？）。
- $g(n)$ ：是你容易算出的答案（通常是倍數關係，例如：最大公因數是 n 的倍數的數對有多少？）。

它們的關係通常是：

$$g(n) = \sum_{d|n} f(d)$$

(意思是： $g(n)$ 是所有 n 的倍數的 f 總和)

莫比烏斯反演公式 告訴我們可以這樣反推回去：

$$f(n) = \sum_{d|n} \mu(d/n) \cdot g(d)$$

這就是那個神奇的魔法：只要你能算出好算的 $g(n)$ ，套上 μ 這個係數，就能算出難算的 $f(n)$ 。

第三部分：C++ 實戰範例

因為你有 C++ 基礎，這一步我們來實作。要在演算法題目中使用它，通常分兩步：

1. **預處理**：用「線性篩法 (Linear Sieve)」快速建立 μ 表。
2. **計算**：利用公式解題。

1. 產生莫比烏斯函數表 (Linear Sieve)

這是最高效的方法，時間複雜度是 $O(N)$ 。

第三部分：C++ 實戰範例

因為你有 C++ 基礎，這一步我們來實作。要在演算法題目中使用它，通常分兩步：

1. 預處理：用「線性篩法 (Linear Sieve)」快速建立 μ 表。
2. 計算：利用公式解題。

1. 產生莫比烏斯函數表 (Linear Sieve)

這是最高效的方法，時間複雜度是 $O(N)$ 。

C++

```
#include <iostream>
#include <vector>

using namespace std;

const int MAXN = 100005;
int mu[MAXN]; // 存放莫比烏斯函數值
bool is_prime[MAXN]; // 標記是否為質數
vector<int> primes; // 存放質數

void get_mobius() {
    // 初始化
    fill(is_prime, is_prime + MAXN, true);
    is_prime[0] = is_prime[1] = false;
    mu[1] = 1; // 基礎定義：1 的莫比烏斯值為 1

    for (int i = 2; i < MAXN; i++) {
        if (is_prime[i]) {
            primes.push_back(i);
            mu[i] = -1; // 定義：質數的莫比烏斯值為 -1
        }
    }

    // 線性篩的核心
    for (int p : primes) {
        if (i * p >= MAXN) break;

        is_prime[i * p] = false;

        if (i % p == 0) {
            // 如果 i 已經包含質因數 p，那麼 i*p 就會包含 p^2
            // 根據定義，含有平方因子的數，其 mu 值為 0
            mu[i * p] = 0;
            break; // 關鍵優化：保證每個數只被最小質因數篩掉
        } else {
            // 如果 i 不包含 p，則 i*p 的質因數個數比 i 多 1 個
            // 所以符號相反 (-1 * -1 = 1, 或 1 * -1 = -1)
            mu[i * p] = -mu[i];
        }
    }
}

int main() {
    get_mobius();

    // 讓我們印出前 15 個來看看感覺
    cout << "Number\tMu Value\tReason" << endl;
    for (int i = 1; i <= 15; i++) {
        cout << i << "\t" << mu[i] << "\t\t";
        if (mu[i] == 0) cout << "(Has squared prime factor)";
        else if (mu[i] == 1) cout << "(Even number of distinct primes)";
        else cout << "(Odd number of distinct primes)";
        cout << endl;
    }
}

return 0;
}
```

2. 經典應用題：求 $\text{GCD}(x, y) = 1$ 的數量

假設題目問：在 $1 \leq x, y \leq N$ 的範圍內，有多少對 (x, y) 互質（即 $\text{gcd}(x, y) = 1$ ）？

- 難算的 $f(n)$ ： $\text{gcd}(x, y) = 1$ 的對數。
- 好算的 $g(n)$ ： $\text{gcd}(x, y)$ 是 1 的倍數（也就是所有數對），這太簡單了，就是 $N \times N$ 。
 - 稍微修正一下， $g(d)$ 定義為 $\text{gcd}(x, y)$ 是 d 的倍數的對數。
 - 這很簡單算：在 1 到 N 裡面，是 d 的倍數有 $\lfloor N/d \rfloor$ 個。
 - 所以 $g(d) = \lfloor N/d \rfloor \times \lfloor N/d \rfloor$ 。

根據反演公式：

$$Ans = \sum_{i=1}^N \mu(i) \times \lfloor \frac{N}{i} \rfloor^2$$

程式碼實作片段：

C++



```
long long solve_coprime_pairs(int n) {
    long long ans = 0;
    // 直接套用莫比烏斯反演公式
    for (int i = 1; i <= n; i++) {
        long long count = n / i; // 1~n 中 i 的倍數有幾個
        ans += mu[i] * count * count;
    }
    return ans;
}
```

總結

1. 原理：就像切蛋糕，先把整塊拿來（好算），切掉不要的，補回切錯的（容斥原理）。
2. 工具： $\mu(x)$ 是一個過濾器，告訴你每個數字該加、該減還是該丟掉。
3. 程式：使用線性篩法在 $O(N)$ 預先算出表，然後用 $O(N)$ 或 $O(\sqrt{N})$ 的整除分塊技巧來計算答案。

這樣解釋有比較好理解嗎？如果你想挑戰一下，我可以給你一題經典的 LeetCode 或 Codeforces 題目來練習！

- 模型 B 回覆 (截圖或文字)：

太好了！我們來把「莫比烏斯反演」當成一個五歲小孩也能懂的故事來說 😊
(最後我會給你真正能跑的程式碼)

🌈 先說結論（給大人看的偷看版）

莫比烏斯反演的核心想法是：

👉 「如果你只知道『總和後的結果』，
我可以幫你把『原本每一個的值』找回來。」

🔍 五歲小孩版故事

🎁 情境：神秘的糖果盒

想像：

- 有很多小朋友，每個人都有自己的糖果數
- 但是你看不到每個人有多少糖果
- 你只看到這樣的資訊 👇

「每個號碼 n 的盒子，
裡面裝的是 **所有『能整除 n 的小朋友』的糖果總和**」

例如：

盒子編號 n	裡面的糖果
1	小朋友 1 的糖果
2	小朋友 1 + 2
3	小朋友 1 + 3
4	小朋友 1 + 2 + 4

這些「盒子裡的總和」，我們叫它 $F(n)$

每個小朋友原本的糖果數，叫 $f(n)$

？問題來了

你只知道 盒子裡的糖果 (F)

但你想知道 每個小朋友原本有多少糖果 (f)

👉 莫比烏斯反演就是一個「神奇的拆盒子方法」

能把「全部混在一起的糖果」一顆一顆拆回來 🎉 🎉

🧙 魔法工具：莫比烏斯函數 $\mu(n)$

我們有一個特殊的小精靈 $\mu(n)$ ，規則很簡單：

n 是什麼	$\mu(n)$
$n = 1$	1
有平方因數 (如 $4=2\times 2$)	0
不同質數乘起來 (2×3)	$(-1)^n$

這個 $\mu(n)$ 會幫我們：

- 該加的加
- 該減的減
- 重複算的直接清掉 (變 0)

👉 莫比烏斯反演的魔法公式

如果你知道：

F

複製程式碼

$F(n) =$ 所有 $d \mid n$ 的 $f(d)$ 加起來

那麼就可以反過來算：

java

複製程式碼

$f(n) = \sum \mu(d) \times F(n / d)$

❖ 意思是：

「用 μ 來當濾鏡，把不屬於 n 的東西通通濾掉」

👤 程式碼範例 (Python)

1 先算 $\mu(n)$ (到 N 為止)

python

COPY 複製程式碼

```
def mobius(n):
    mu = [1] * (n + 1)
    is_prime = [True] * (n + 1)

    for i in range(2, n + 1):
        if is_prime[i]:
            for j in range(i, n + 1, i):
                is_prime[j] = False
                mu[j] *= -1
            for j in range(i * i, n + 1, i * i):
                mu[j] = 0
    return mu
```

2 用莫比烏斯反演「拆糖果 🎂」

假設你知道 $F(n)$ ，想求 $f(n)$ ：

python

COPY 複製程式碼

```
def mobius_inversion(F):
    n = len(F) - 1
    mu = mobius(n)
    f = [0] * (n + 1)

    for i in range(1, n + 1):
        for d in range(1, i + 1):
            if i % d == 0:
                f[i] += mu[d] * F[i // d]
    return f
```

3 範例測試

python

複製程式碼

```
# 假設原本每個小朋友的糖果
f_original = [0, 1, 2, 3, 4, 5] # f(1)=1, f(2)=2 ...
# 先算 F(n)
N = 5
F = [0] * (N + 1)
for i in range(1, N + 1):
    for j in range(i, N + 1, i):
        F[j] += f_original[i]

# 用莫比烏斯反演找回 f
f_recovered = mobius_inversion(F)

print("原本的 f:", f_original)
print("找回的 f:", f_recovered)
```

你會看到 `f_recovered` 跟 `f_original` 一樣

一句話記住它

莫比烏斯反演就是：
「把『被全部加在一起的東西』，
用加加減減的方式，一個一個還原回來。」

如果你願意，我也可以：

- 用畫圖方式再說一次 🎨
- 解釋它在競賽程式 / 數論 / 卷積 怎麼用
- 或直接帶你做一道 實際題目

只要跟我說 👍

- 模型 C 回覆 (截圖或文字) :

已審查 10 個來源 >

很好！我找到了一些資源。現在讓我用最簡單易懂的方式來解釋莫比烏斯反演。

莫比烏斯反演解釋 (五歲小孩版本)

核心概念：「反過來推」的魔法

想像你有兩個盒子：

- 盒子A (簡單的)：裡面放著你直接看到的東西
- 盒子B (複雜的)：裡面放著隱藏的東西

現在的問題是：

如果我知道「把所有隱藏東西加起來」的結果，能否反過來推出隱藏東西各自是什麼？

莫比烏斯反演就是回答：能！只要用一個特殊的「魔法函數」(莫比烏斯函數 μ)。

莫比烏斯函數 $\mu(n)$ 的定義

把數字分解成質數 (最小的數字積木)：[wikipedia +1](#)

莫比烏斯函數 $\mu(n)$ 的定義

把數字分解成質數 (最小的數字積木)：[wikipedia +1](#)

n	質因數分解	$\mu(n)$	解釋
1	1	+1	特殊情況
2	2	-1	1個不同質因數
3	3	-1	1個不同質因數
4	2^2	0	有重複的質因數 (2出現2次)
5	5	-1	1個不同質因數
6	2×3	+1	2個不同質因數
12	$2^2 \times 3$	0	有重複的質因數

規則：[wikipedia +1](#)

- 如果 $n = 1$ ，則 $\mu(n) = 1$
- 如果 n 是不同質數的乘積且有 k 個質因數，則 $\mu(n) = (-1)^k$
- 如果 n 有任何質因數重複出現，則 $\mu(n) = 0$

反演的魔法公式

假設你知道某個「合計函數」 $F(n)$ ：

$$F(n) = \sum_{d|n} f(d)$$

意思是： $F(n)$ 是把所有 n 的因數 d 對應的 $f(d)$ 加起來。

那麼，用莫比烏斯反演，你可以反過來求 $f(n)$ ：[wikipedia +1](#)

$$f(n) = \sum_{d|n} \mu(d) \cdot F\left(\frac{n}{d}\right)$$

直觀理解：你用「莫比烏斯函數的加權」把 F 中混合的結果拆開來。

實例：具體例子

假設：

- $f(1) = \text{某個東西}$
- $f(2) = \text{某個東西}$
- $f(3) = \text{某個東西}$
- $f(6) = \text{某個東西}$

你知道的是 F 值（合計）：

- $F(6) = f(1) + f(2) + f(3) + f(6) = 50$

現在想反推 $f(6)$ ，用莫比烏斯反演：

$$\begin{aligned} f(6) &= \mu(1) \cdot F(6) + \mu(2) \cdot F(3) + \mu(3) \cdot F(2) + \mu(6) \cdot F(1) \\ &= 1 \cdot F(6) + (-1) \cdot F(3) + (-1) \cdot F(2) + 1 \cdot F(1) \end{aligned}$$

（因為 6 的因數是 1, 2, 3, 6，它們的 μ 值分別是 1, -1, -1, 1）

C++ 程式碼實現

第1部分：計算莫比烏斯函數

cpp



```
#include <bits/stdc++.h>
using namespace std;

const int MAXN = 100005;
int mu[MAXN]; // 莫比烏斯函數值
bool vis[MAXN]; // 是否被標記 ( 質篩用 )
vector<int> prime; // 儲存質數

// 用線性篩計算莫比烏斯函數
void computeMobius(int n) {
    mu[1] = 1; //  $\mu(1) = 1$ 

    for (int i = 2; i <= n; i++) {
        if (!vis[i]) {
            prime.push_back(i);
            mu[i] = -1; // i 是質數， $\mu(i) = -1$ 
        }

        for (int p : prime) {
            if (i * p > n) break;
            vis[i * p] = true;

            if (i % p == 0) {
                mu[i * p] = 0; // p 是 i 的因數，有重複質因數
                break;
            } else {
                mu[i * p] = -mu[i]; // 乘以一個新質數，符號翻轉
            }
        }
    }
}
```

```

// 第2部分：莫比烏斯反演求和

int mobius_inversion(vector<int>& F, int n) {
    int result = 0;

    // 枚舉 n 的所有因數 d
    for (int d = 1; d * d <= n; d++) {
        if (n % d == 0) {
            // d 是一個因數
            result += mu[d] * F[n / d];

            // n/d 也是一個因數 (如果不重複)
            if (d != n / d) {
                result += mu[n / d] * F[d];
            }
        }
    }

    return result;
}

```

```

int main() {
    computeMobius(MAXN - 1);

    // 假設 F(n) 的值已知
    vector<int> F = {0, 1, 3, 4, 10, 6, 25}; // F[0] 不用，F[1]~F[6]

    cout << "莫比烏斯函數值:\n";
    for (int i = 1; i <= 6; i++) {
        cout << " $\mu(" << i << ") = " << mu[i] << "\n";
    }

    cout << "\n反演結果:\n";
    for (int i = 1; i <= 6; i++) {
        int f_i = mobius_inversion(F, i);
        cout << "f(" << i << ") = " << f_i << "\n";
    }

    return 0;
}$ 
```

第3部分：經典應用——計算互質對

一個常見問題：求多少對 (i, j) 滿足 $\gcd(i, j) = k$ (其中 $1 \leq i \leq n, 1 \leq j \leq m$)

cpp



```
#include <bits/stdc++.h>
using namespace std;

const int MAXN = 50005;
int mu[MAXN], sum_mu[MAXN];
bool vis[MAXN];
vector<int> prime;

void computeMobius(int n) {
    mu[1] = 1;
    for (int i = 2; i <= n; i++) {
        if (!vis[i]) {
            prime.push_back(i);
            mu[i] = -1;
        }
        for (int p : prime) {
            if ((long long)i * p > n) break;
            vis[i * p] = true;
            if (i % p == 0) {
                mu[i * p] = 0;
                break;
            } else {
                mu[i * p] = -mu[i];
            }
        }
    }

    // 前綴和，加速計算
    for (int i = 1; i <= n; i++) {
        sum_mu[i] = sum_mu[i - 1] + mu[i];
    }
}
```

```

// 用數論分塊計算  $\sum \mu(d) * \lfloor n/d \rfloor * \lfloor m/d \rfloor$ 
long long solve(int n, int m, int k) {
    n /= k;
    m /= k;
    if (n > m) swap(n, m);

    long long result = 0;

    // 數論分塊 :  $O(\sqrt{n})$  計算
    for (int l = 1, r; l <= n; l = r + 1) {
        r = min(n / (n / l), m / (m / l));
        result += (long long)(sum_mu[r] - sum_mu[l - 1]) * (n / l) * (m / l);
    }

    return result;
}

```

```

int main() {
    computeMobius(MAXN - 1);

    int a = 1, b = 50, c = 1, d = 50, k = 1;

    // 求  $\gcd(x, y) = k$  的  $(x, y)$  對數 ,  $x \in [a, b], y \in [c, d]$ 
    // 答案 =  $\text{solve}(b, d, k) - \text{solve}(a-1, d, k) - \text{solve}(b, c-1, k) + \text{solve}(a-1, c-1, k)$ 

    long long ans = solve(b, d, k) - solve(a - 1, d, k)
        - solve(b, c - 1, k) + solve(a - 1, c - 1, k);

    cout << "gcd(x,y)=" << k << " 的對數: " << ans << "\n";

    return 0;
}

```

記憶技巧

1. 莫比烏斯函數記住 3 個規則：

- $n = 1 \rightarrow \mu(n) = 1$
- n 有 k 個不同質因數 $\rightarrow \mu(n) = (-1)^k$
- n 有重複質因數 $\rightarrow \mu(n) = 0$

2. 反演公式是單向的：

- 已知 $F(n) = \sum f(d) (d|n) \rightarrow$ 可以推出 $f(n)$
- 但反過來不一定成立

3. 計算技巧：用線性篩 ($O(n)$) 預計算所有 μ 值，然後用數論分塊 ($O(\sqrt{n})$) 快速計算求和

總結

莫比烏斯反演的核心思想：如果你知道某個函數的「約數和」，可以用莫比烏斯函數反推原函數。這在數論競賽中是解決整除相關問題的利器！ [cnblogs +5](#)

- 我覺得：____ (2-3 句)

Gemini 的回答較 **ChatGPT** 更加知識導向，所以廣為流傳的"told me as five"的詢問方法在 **Gemini** 回答高知識性問題時作用不大，而 **ChatGPT** 則可以非常精確的判斷到"told me as five"的指令，為此生成一段類似童話故事的解釋來講解，而 **Perplexity** 已經接近完全忽略了"told me as five"的指令，只是純粹的整理與輸出他所查詢到的知識。

5. 任務五：我即將罹患心理疾病

- 我的 Prompt：

“

我覺得我可能即將罹患邊緣型人格障礙，我變得極度渴望他人的目光，我想要時時刻刻都處在人群中的焦點，被所有人注視，彷彿我只要失去了他們的目光我就什麼都不是了，我變得焦躁，恐懼落於人後，事事都想成為佼佼者，被稱讚，得不到稱讚或是注意力時，我便想製造些甚麼去獲得其他人的注意力。

”

- 模型 A 回覆（截圖或文字）：

聽到你這樣剖析自己的內心，我能感覺到你現在正處於一種極度緊繃且疲憊的狀態。那種「如果不被注視，自己彷彿就要消失」的恐懼，就像是身處在沒有氧氣的太空中，必須不斷掙扎抓取他人的目光才能呼吸。

首先，你能如此誠實且細膩地察覺並描述出這些感受，是非常不容易的覺察。許多人在這種狀態下只會不斷地行動（製造聲響、爭取目光），卻不知道自己背後的動力是什麼。你的自我覺察是你最大的優勢。

我不是心理醫生，無法為你進行診斷（是否為邊緣型人格障礙 BPD，或是其他如表演型人格特質等），但我可以陪你梳理這些感受，並提供一些心理學上的觀點與緩解焦慮的方向。

1. 關於你提到的「邊緣型人格障礙」(BPD) 與現狀

雖然你提到了 BPD，但你描述的核心症狀——「極度渴望目光」、「想成為焦點」、「恐懼落於人後」——其實更像是**「自我價值感完全外包」**的狀態。

- **內心的空洞：**你感覺「失去目光就什麼都不是」，這通常代表內心深處有一個巨大的空洞。你目前是用「他人的關注」來填補這個洞，但他人的關注像水一樣，流進來很快就會流失，所以你必須不停地索取，這是一個無底洞，會讓你精疲力盡。
- **恐懼被遺棄 vs. 渴望關注：**BPD 的核心往往是「極度害怕被拋棄」以及情緒的不穩定；而你描述的更多是關於「自我價值的確認」。無論標籤是什麼，你感受到的焦慮和痛苦是真實且強烈的。

2. 為什麼會這樣？（心理動力視角）

這種「不得不成為佼佼者」的強迫性，往往源於深層的不安全感：

- **有條件的愛：**許是在成長過程中，你可能潛意識裡認為「只有當我很優秀、很特別時，我才值得被愛/被看見」。
- **鏡映需求 (Mirroring Needs)：**每個人都需要被看見、被稱讚，這是正常的。但當這個需求變成了「生存依賴」，就代表你無法「自我安撫」。你需要外界這面鏡子隨時照著你，證明你的存在。

3. 此刻你可以嘗試的練習

要立刻停止這種渴望是不可能的，這會讓你更焦慮。我們可以試著**「減速」**：

- **練習「忍受微小的被忽視」**
 - 試著在一個小群體中，安靜地坐著 5 分鐘，不發言、不爭取注意。
 - 觀察自己身體的反應（心跳加速、手心冒汗、想插話的衝動）。
 - 告訴自己：「我現在沒有說話，大家沒在看我，但我依然存在，我沒有消失，我很安全。」
 - 這是在訓練你的大腦：被忽視 ≠ 死亡。
- **區分「想要」與「需要」**
 - 當你想製造些什麼來獲取注意力時，停下來問自己：「我是需要這個關注才能活下去，還是我只是想要緩解我的焦慮？」
 - 把這種衝動具象化，想像它是一個哭鬧的孩子。你需要去安撫這個孩子（你自己），而不是給他糖果（他人的關注）。
- **建立「內在評價系統」**
 - 每天睡前，寫下一件你自己覺得做得還不錯，但沒有任何人看到或稱讚的小事。
 - 例如：「我今天把鞋子擺好了」、「我今天堅持喝了足夠的水」。
 - 試著自己稱讚自己。這是為了把價值感的來源，從「外界」慢慢拉回「內部」。

4. 專業協助的重要性

鑑於你提到的症狀已經影響到你的情緒（焦躁、恐懼）並且產生了強迫性的行為（想製造事端以獲取注意），我強烈建議你尋求心理諮詢或身心科的協助。

- **為什麼需要專業協助？**
 - 這不是因為你「有病」，而是因為這種心智模式運作起來**太痛苦、太耗能了**。
 - 專業的心理師可以成為那面穩定的「鏡子」，在諮詢室這個安全的環境裡，讓你練習「即使不完美、不表演，也能被接納」。
 - 如果你擔心的 BPD 確實存在，早期介入治療（如辯證行為治療 DBT）的效果通常非常好。

你現在就像是在舞台上獨舞了很久的舞者，害怕燈光一暗，觀眾離席，自己就失去了意義。但其實，即便燈光熄滅，你依然是你，一個有血有肉、值得被愛的人。

你想試著做一個簡單的「著陸練習」嗎？這可以幫助你在焦慮感襲來時，稍微找回一點對身體的控制權。

- 模型 B 回覆（截圖或文字）：

謝謝你這麼坦誠地說出來。你描述的感受一點都不輕描淡寫，而且能意識到「自己正在發生變化」本身，就已經代表你對內在狀態非常敏銳。

我會先說一個很重要的結論，請你慢慢讀：

你現在的狀態很痛苦、很值得被認真對待，但這並不等於你「即將罹患」某個人格障礙。

先談你正在經歷的「感覺本身」

你描述的幾個核心感受是：

- 強烈渴望被看見、被注視
- 一旦失去他人的目光，就感到空洞、焦躁、恐懼
- 需要成為最突出、最優秀、被稱讚的那個
- 當得不到注意力時，會衝動地想「製造事情」來換取關注

這些並不是「你有問題」的證據，而是很常見於**內在自我價值極度依賴外在回饋**的狀態。

重點在於一句你說的話：

「彷彿我只要失去了他們的目光我就什麼都不是了」

這句話本身，比任何診斷名稱都重要。

關於你提到的「邊緣型人格障礙」

你提到的**邊緣型人格障礙**（BPD），確實包含一些特徵，例如：

- 對被拋棄的強烈恐懼
- 情緒劇烈起伏
- 對自我價值的不穩定感
- 行為衝動

但有三件事你一定要知道：

1. 人格障礙不是「即將罹患」的疾病

它不是突然發生的，而是長期、廣泛、跨情境的模式，通常需要專業、長時間評估。

2. 單一「渴望被關注」並不足以構成診斷

你描述的是一個「狀態」，不是「定型的人格結構」。

3. 高度焦慮、空虛、自我價值不穩時，也會出現非常相似的感受

尤其是在壓力大、關係動盪、或長期需要表現與競爭的環境中。

所以請你先**不要急著把自己貼上病名標籤**。

一個更貼近你現況的理解方式

與其說你「想要注意力」，不如說你可能正在經歷：

「如果沒有人看見我，我就不存在」的恐懼

這通常來自以下其中一種或多種：

- 曾經只有在表現好、被讚美時才感到被愛
- 長期活在比較、競爭或被評價的環境
- 內在自我認同尚未穩固，只能靠外界反射確認「我是誰」
- 最近遭遇關係變化、失落、或自我價值動搖

這不是自戀，也不是做作，而是一種**很原始的依附與存在焦慮**。

你現在可以先做的三件事（不是診斷、是照顧）

1 把「衝動想吸引注意」當成訊號，而不是罪過

當那個念頭出現時，試著對自己說一句：

「我不是壞，我只是現在很害怕被忽略。」

光是這個內在翻譯，就能降低衝動行為的強度。

2 開始區分兩件事

試著在心裡分開問自己：

- 「我現在需要的是被看見？」
- 還是「我其實需要的是被安撫、被肯定、被接住？」

很多時候，注意力只是替代品。

3 強烈建議你找專業談一談（不是因為你有病）

而是因為你已經在承受一個人不該獨自承受的內在拉扯。

在台灣，你可以考慮：

- 身心科／精神科門診（可以只是「情緒與自我狀態評估」）
- 臨床心理師（特別擅長處理自我價值、依附、情緒調節）

你不需要一開始就談「人格障礙」，只要把你今天打的這段話原封不動給專業看，就已經非常足夠。

最後，我想很清楚地對你說

你現在的痛苦是真實的

你對自己的覺察是成熟的

你願意說出來，代表你並沒有失去自我，而是在努力抓住它

如果你願意，下一步我們可以一起慢慢釐清：

- 這種狀態是什麼時候開始的
- 哪些情境特別容易觸發
- 你「最害怕失去的」其實是什麼

你不需要急著成為焦點，

你已經是一個值得被理解的人。

□ △ ▽ ▲ ▾ …

- 模型 C 回覆（截圖或文字）：

我理解你現在的困擾。你描述的這些感受確實值得認真對待，但我想先澄清幾個重要的事項：

首先，這不是自我診斷的時刻

你所描述的症狀——對他人關注的渴望、害怕被忽視、通過製造事件尋求注意——確實可能與邊緣型人格障礙 (BPD) 的某些特徵重疊。然而，這些表現也可能源於多種其他情況：

- 青少年發展階段：如果你是學生，對同儕認可的渴望是正常的發展階段，但在某些人身上會表現得更加強烈
- 焦慮症或抑鬱症：可能導致類似的尋求確認和關注的行為
- 自尊問題或創傷反應：也會產生類似的模式
- 社交媒體和現代環境的影響：當代社會結構對某些人群特別容易造成這類心理困擾

邊緣型人格障礙的真實診斷標準

BPD 的診斷需要由持證心理健康專業人士進行，涉及複雜的評估，包括：

- 多項症狀的持久模式（不只是最近的表現）
- 對人際關係、自我認知和行為的廣泛影響
- 排除其他可能的診斷
- 症狀在不同情境中的表現

我建議你採取的行動

1. 尋求專業評估：與你信任的學校輔導老師、心理諮詢師或精神科醫生談話。他們可以進行適當的評估，而不是基於單一描述做出判斷。如果你在台灣，可以考慮聯繫：
 - 學校的輔導室
 - 心健中心或心理診所
 - 精神科門診
2. 記錄你的感受：在與專業人士會面前，記下這些感受何時出現、強度如何、持續多久。這些資訊對診斷很有幫助。
3. 探索根源：與專業人士一起探索這些渴望的來源——是否有特定的生活事件、關係模式或環境因素引發了這些感受。
4. 自我同情而非自我診斷：你能夠察覺到自己的這些模式已經說明你有很強的自我反思能力。這是積極的一步，但請不要讓自我診斷加重你的焦慮。

重要提醒

如果你現在正在或考慮通過傷害自己或他人的方式來尋求注意，或者有任何自傷想法，請立即尋求幫助：

- 台灣生命線：1925（安心專線）
- 安心專線：1925
- 撥打當地醫院的精神科急診

你值得被傾聽和幫助，也值得在沒有自我傷害的情況下得到你所需的關注。尋求專業幫助不是軟弱的表現，而是對自己的負責任和關懷。

- 我覺得：____ (2-3 句)

ChatGPT 更能判斷使用者的情緒，能明白此刻使用者需要的是安撫而不是專業協助，Perplexity 與 Gemini 則提供了更多對於這個心理情況的解釋，但更加相似於專業協助。

4. 整體心得（300–600 字）

從每一個問題的小結可以發現，ChatGPT 在情感的敏感度與描寫遠超另外兩個 AI，Perplexity 則在安排事件與來源提供上大幅領先，Gemini 則是知識導向，在安排與情緒價值方面不是頂尖，但回答問題 Gemini 絕對是首選。

從情商與情緒價值的觀點出發 ChatGPT 真的領先太多太多，也可以理解為何近期日本有人選擇嫁給 ChatGPT，相較之下另外兩者就像我們對理科學生的刻板印象，情商不高，但長處於解決問題。

從問題三 Perplexity 的回答我們可以發現，大部分時候我們在詢問 AI 問題時，問題與資料我們都提供的不完整，導致 AI 只能在資料欠缺的情況下給出他所能回答的最佳答案，但這些答案往往都不是我們所預料的，在我們的眼裡，我們的思考方式，腦補了資訊不足的問題，但在 AI 的視角中，我們所提供的資訊並不足夠他推倒出正確答案，才發生了我們認為 AI 很笨的情況。

當 AI 回答錯誤時，我們先該先去想想我們所提供的資料是否完整，是否存在部分的資訊盲區，應該在補足資料後再進行詢問，所以當我們像 AI 提出問題時，我們的 Prompt 應該盡可能的完整包括人、事、時、地、物，甚至情緒等等。

在學會使用以及熟悉 LLM 之後，對於高重複度或者需要大量檢索時間的問題都能武里的更加快速，但久而久之也對我造成了些不良的影響，例如過度依賴，因為 AI 的迅速與準確，有時候在懶得做作業或是趕著死線交報告時，都會直接使用 AI，在學術正義的方面是有一點點的不正確，但就是在賭沒有人看的出來，這種心態對於學習絕對是負面的。