

2025/5/14

「作業系統」作業四 (Due 2025/4/30(三)13:00)

請於截止時間前把完成的程式碼或把整個專案(但**移除所有執行檔**)壓縮成單一檔案，主檔名設為 OS4 再透過 <https://reurl.cc/M37Mzp> 上傳繳交，上傳頁面的「學號欄」務必填寫以利識別繳交資料的同學，謝謝。

「模擬退火法」(simulated annealing)是紀老師推薦同學熟悉的搜尋方法，它對於解的可能性繁多之最佳化問題十分有用，原理不難但效果顯著，也經常應用於人工智慧領域，值得學習。本次作業將練習以多執行緒實作模擬退火法解「旅行銷售員問題」(travelling salesman problem)。具備「多執行緒程式化」的觀念與技巧，對未來解決實務問題很有幫助。

設有八個城市編號分別為 1, 2, ..., 8，彼此之間的交通距離成本以矩陣表示如下：

Distances between each pair of cities (unit: km)								
	1	2	3	4	5	6	7	8
1	—	91.8	105.2	89.9	189.9	76.2	278.3	54.4
2	91.8	—	187.2	38.9	271.3	162.9	363.3	88.4
3	105.2	187.2	—	194.1	182.3	31.4	176.1	153.8
4	89.9	38.9	194.1	—	249.4	166.1	368.3	63.6
5	189.9	271.3	182.3	249.4	—	168.0	243.0	185.9
6	76.2	162.9	31.4	166.1	168.0	—	202.2	122.8
7	278.3	363.3	176.1	368.3	243.0	202.2	—	320.0
8	54.4	88.4	153.8	63.6	185.9	122.8	320.0	—

請撰寫一份 C 程式內有三個執行緒，每一個執行緒實作使用相同參數(初始溫度 T_1 設為 100、降溫率 α 設 0.95、降溫最多 N 次設 100)的模擬退火法，程式將綜合三個執行緒所找到的最佳路徑，再從中取最優者作為程式最後的輸出。程式畫面如下，但請留意：模擬退火法內含隨機因素，每次所得結果不盡相同，同學的輸出畫面也很可能與底下有出入，只要最終結果(最佳路徑與最低成本)與圖例一致即可放心。

*** 多執行緒實作模擬退火法解旅行銷售員問題 ****

創建3個執行緒運算過程摘要如下 ...

執行緒0發現較佳路徑，成本=1360.5

執行緒0發現較佳路徑，成本=1322.2

執行緒0發現較佳路徑，成本=1265.2

執行緒0發現較佳路徑，成本=1237.2

執行緒0發現較佳路徑，成本=1064.8

執行緒1發現較佳路徑，成本=1411.3

執行緒1發現較佳路徑，成本=1377.2

執行緒1發現較佳路徑，成本=957.8

執行緒0發現較佳路徑，成本=1007.4

執行緒0發現較佳路徑，成本=906.9

執行緒2發現較佳路徑，成本=1360.5

執行緒2發現較佳路徑，成本=1151.8

執行緒2發現較佳路徑，成本=1097.7

執行緒2發現較佳路徑，成本=1041.7

執行緒2發現較佳路徑，成本=962

執行緒2發現較佳路徑，成本=906.9

多執行緒協力找得最佳路徑：1→6→3→7→5→8→4→2→1；最低成本=906.9（歸功執行緒0運算）
參見thread0-output.txt, ..., thread2-output.txt檔案載有各個執行緒求解過程細部資訊

建議同學研讀課堂網頁公告此次作業的相關素材(一份 [PPT 檔](#)、一份[單執行緒程式碼](#)、一份[多執行緒程式碼](#))，將有助於完成作業。以上為程式要求達成的功能，其餘未規範的項目可自行設計。