

Section 1. 如何實現 P1

1. `lexicographical(path)` 函数、`next_permutation(path)` 函数：

這兩函数用字典序算法來窮舉出所有可能排列。

Step1 找到最大的索引 k ，使得 $path[k] < path[k+1]$ 。

Step2 找到最大的索引 l ，使得 $path[l] > path[k]$ 。

Step3 交換 $path[k]$ 和 $path[l]$ 。

Step4 將 $path[k+1:]$ 反轉確保排列按字典序排列。

Step5 返回下一個排列。

2. `total_dist(path, mat)` 函数：

這個函数用於計算給定路徑的總路徑長度。

Step1 在路徑的開始和結束插入城市 0，確保從城市 0 出發回到城市 0。

Step2 遍歷路徑上的城市，計算城市之間的距離。

Step3 返回總路徑長度。

3. `BF(mat)` 函数：

這是 Brute-Force Search 的主要函数。

Step1 初始化一個最佳路徑 (`best_path`)。

Step2 計算窮舉可能的排列數 (`exhaustive_num`)。

Step3 創建一個初始路徑 `path`，它包含城市 1 到城市 `city_num-1`。

Step4 在一個循環中，計算 `path` 的總路徑長度 `current_total_dist`。

Step5 然後使用字典序演算法生成下一個排列。

Step6 如果當前路徑更短，則更新最佳路徑和最佳距離，

Step7 循環直到所有可能的排列都被考慮過。

Section 2. 如何實現 P2

此題我利用上課講義提供的 pseudocode 完成

1. **total_dist(path, mat)** 函數：與 P1 功能相同
2. **SA(mat, init_path, init_temp, cooling_rate, num_iter)** 函數：

SA 算法的主要函數，接受以下參數：

- **mat**：城市之間的距離矩陣。
- **init_path**：初始路徑，不包括城市 0。
- **init_temp**：初始溫度。
- **cooling_rate**：冷卻率。
- **num_iter**：迭代次數。

運算步驟如下：

Step1 初始化當前路徑和距離，設置初始溫度。

Step2 開始迭代，每次迭代中：

- 降低溫度。
- 隨機選擇兩個不同的城市進行交換。
- 計算能量變化（距離的變化）。
- 如果新路徑較短，則接受新路徑。
- 根據一定的機率接受較差的路徑，允許跳出局部最小值。

Step3 迭代結束後，返回最佳路徑和最佳距離。

Section 3.

1. **路徑優化過程**：初始時，模擬退火算法可能會選擇較差的路徑，但隨著溫度的逐漸降低，算法會趨向於選擇更優的路徑。
2. **溫度下降**：改變初始溫度和冷卻率會影響模擬退火的溫度下降速度。較高的初始溫度和較慢的冷卻率會導致更多的隨機性和更長的搜索時間，而較低的初始溫度和較快的冷卻率會導致更局部的搜索。
3. **局部最優解**：模擬退火算法是一種全局搜索方法，可以避免陷入局部最優解

Section4.

Brute Force (BF)：

優點：

1. **精確性**：BF 不依賴於隨機性，可以找到確切的最優。
2. **理解和實現簡單**：BF 的實現相對簡單，且易於理解。

缺點：

1. **計算需求**：BF 需要計算大量的可能路徑，這將需要非常長的時間。BF 在 NP-hard 問題上的時間複雜度為 $O(n!)$ ，其中 n 是城市數量。

Simulated Annealing (SA)：

優點：

1. **適用於大型問題**：SA 是一種啟發式算法，適用於處理大量的可能路徑，因為它不需要計算所有可能的路徑。
2. **全局搜索**：SA 能夠逃離局部最優解，因為它以一定的概率接受更差的解，這有助於更好地探索搜索空間。
3. **可調整的參數**：可以通過調整初始溫度、冷卻率等參數來平衡搜索速度和結果品質。

缺點：

1. **不保證最優解**：SA 不像 BF 那樣能夠確保找到最優解，它只能找到一個近似最佳解，該解的品質取決於算法的參數設置。
2. **對參數敏感**：SA 的性能高度依賴於初始溫度、冷卻率等參數的選擇，不同參數可能導致不同的結果。
3. **隨機性**：SA 是一種隨機性算法，多次運行可能產生不同的結果。

心得：

透過這次實作讓我了解上課提供的演算法與 pseudocode，對於解決組合優化問題非常有幫助。在這次的實作中，我學習了兩種不同的算法，Brute Force (BF) 和 Simulated Annealing (SA)，並瞭解了它們的優缺點以及如何實際實現它們。