1. **在課程中我們提到了一些 NP-Complete 問題，包含了 SAT、CNF-SAT、3-CNF-SAT、TRAVEL SALESMAN PROBLEM、HAMILTONIAN PATH、HAMILTONIAN CYCLE、VERTEX COVER、EDGE COVER、CLIQUE、DOMINATING SET、0-1 KNAPSACK。但是 NP-Complete 包含很多問題，請自行選出五個 NP-Complete 問題定義作介紹。你選擇的五個 NP-Complete 問題不得與已經在課程投影片當中出現過。一個值得參考的網頁是 https://en.wikipedia.org/wiki/List_of_NP-complete_problems；裡面涵蓋了蠻多 NP-Complete 問題，可以從上述列表或是自行網路搜尋找出有興趣的 NP-Complete 問題。你僅只需要使用圖文簡介該問題其輸入輸出並讓助教與我可以知道你真的理解該問題定義即可，無須提出解決該問題的演算法。**

(1) 圖著色問題 Graph Coloring Problem

Input: 無向圖 G=(V, E)

Output: G 相鄰的頂點都被塗上不同的顏色

敘述: 該問題是希望用最少的顏色 k 為頂點 V 塗色，並且相鄰兩頂點的顏色不可重複。

(2) 分團覆蓋問題 Clique cover

Input: 無向圖 G=(V, E)

Output: G 可以分成的 cliques

敘述: 該問題目的是求出 G 可以分出幾個 cliques，並且每個 clique 裡的點有哪些。

(3) 三維匹配問題 3-dimensional matching

Input: W, X, Y 三個不相交的集合

Output: 由 W, X, Y 之元素組成的匹配

敘述: 該問題類似於婚姻問題，只不過將二維換到三維，僅僅二維的婚姻問題是 P，可以在多項式時間內解決，但是要讓三方都能盡量滿意的匹配則更複雜，變成了 NP-C 問題。

(4) 工廠排程問題 Job-shop scheduling

Input: 具有不同處理時間的 n 個作業，具有不同處理能力的 m 台機器

Output: 能在最短時間裡完成所有作業之排程

敘述: 每個作業由一組操作組成，這些操作需要按特定順序進行處理，而每個

操作都要在一台特定的機器需要在其上進行處理，並且在同一時間只能處理作業中的其中一個操作，還有一種簡化版本是每個操作都可以在任何機器上進行處理（即假設機器都是相同的），總之目的都是要設計出最有效率的排程。

(5) 精確覆蓋問題 Exact cover

Input: 集合 X

Output: 若干子集 S 的集合 S*，其中兩兩集合無交集，且全集正好為 X

敘述: 要求出由集合 X 裡分出的若干子集 S 之集合 S*，條件上為讓 X 的每個元素只能出現在其中一個 S 裡一次，而且不能有元素沒出現在 S 裡，也就是「兩兩集合無交集，且全集正好為 X」的概念。

2. **SUBSET-SUM PROBLEM 也是一個 NP-Complete 問題，其問題描述如下。給定 s1, s2, ..., sn 共 n 個非負整數與一個目標數 t。SUBSET-SUM PROBLEM 想問是否存在 s1, s2, ..., sn 的 subset 使得其加總恰好為 t。請試圖證明 SUBSET-SUM PROBLEM ≤p KNAPSACK PROBLEM。可能參考的文件有 http://cgm.cs.mcgill.ca/~avis/courses/360/2003/assignments/sol4.pdf。**

Subset Sum problem 與 knapsack problem 這兩者都是在一個集合裡挑選出一個總和不超過某一值 t，差別在於 knapsack problem 除了元素的「總重量(總和)」之外還會考慮元素的「價值」而挑選，使其總價值最大化。而 Subset Sum problem 僅僅只要假設每一元素價值相等，也就是相當於把「價值」這一屬性從元素裡剔除，就可直接使用 knapsack problem 的演算法快速得出答案(總和等於 t 就是 true，總和小於 t 就是 false)