B07502173 陳閔鴻

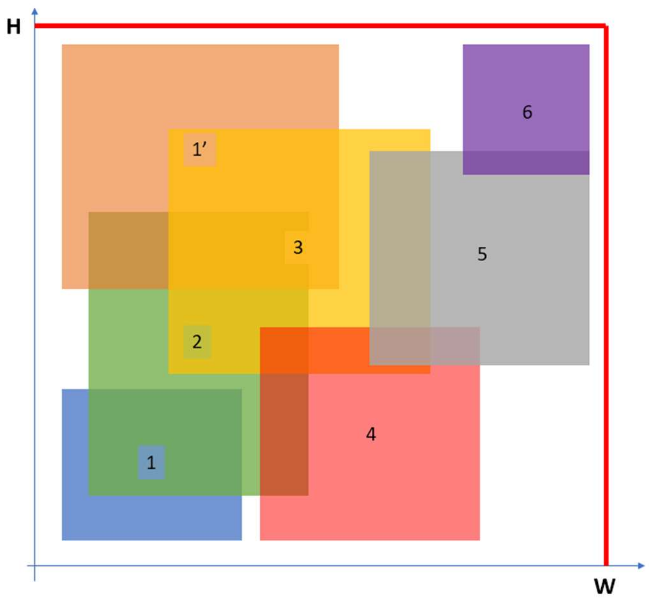
DIY 2

情境發想：

我的阿公務農，一輩子吃苦耐勞，是我極為敬佩的人。他有一片竹林，用以種竹筍維生，而經過這至少一甲子與竹筍的相處，他與阿嬤也將竹筍視為至寶。我有天好奇的向阿公請教種筍的秘訣，才知道有許多眉眉角角。較為重要的幾點是: 種植過程在竹叢周圍需挖溝，以利灌溉; 要提高成活率，栽植時需要調整栽植密度，而竹筍的生長是一群一群的，故可以以一個小區域進行種植與收割。而這「區域」的觀念就激起我的興趣，發現這可以以「方形」作為簡化。若要在一片空曠的山地種植竹筍，並得到最高的收益，可以將其簡化成一個 2D 種植利益最大化的問題。更巧的是，我發現教授講到 Dynamic Programming 的第一個範例，也就是「Weighted Scheduling Problem」或許可以做為參考或問題模型化的方法。於是此 DIY 就這麼誕生了：

問題描述：

Node	Value
1	w_1
1′	$w_{1'}$
2	w_2
3	w_3
4	w_4
5	w_5
6	w_6

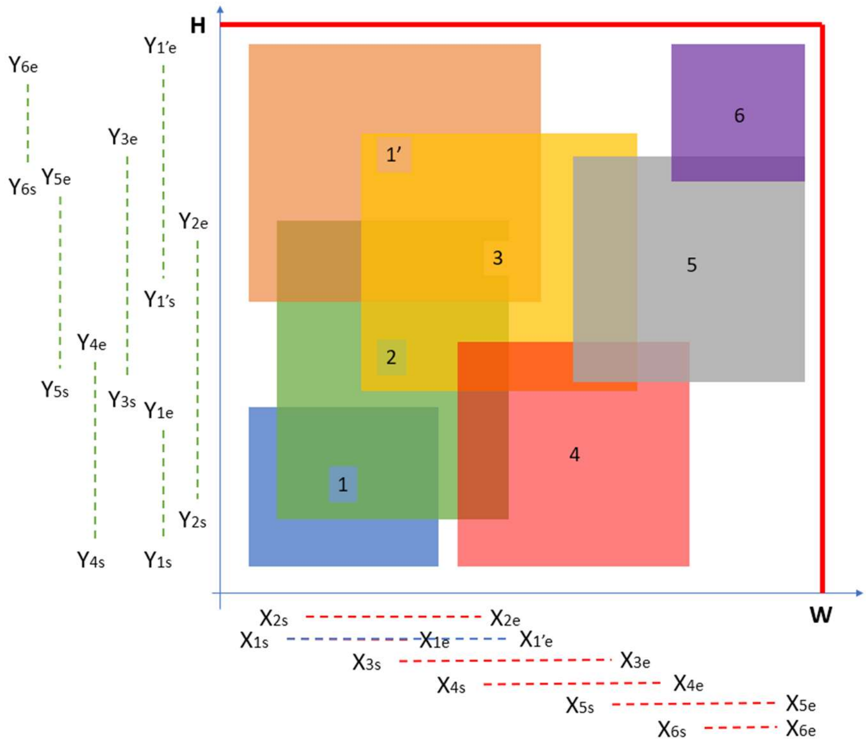


▲ 圖一、問題描述

竹筍種植時會有很多可以種植的候選「區域」，以圖一的各色方塊表示。整個 W*H 的竹園以{1, 1’, 2, 3, 4, 5, 6}這些可能的 subset 組成，而每個 subset 有對應的獲益{w_1, $w_{1'}$, w_2, w_3, w_4, w_5, w_6}，目標是在不重疊區域的狀況下獲得最大收益。

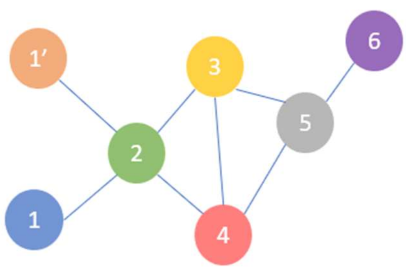
解題思路：

一開始打算想以 2D 的 Weighted Scheduling Problem 作為其模型，如圖二所示，其$\{X_{1s}$、$X_{1e}\}$代表 task 1 的 x 座標的 schedule 範圍，其他同理。但後來發現分兩個維度解完後，兩個維度結果要結合很困難(可能要設計出一個方法)。



▲ 圖二、Modeled as Weighted Scheduling Problem

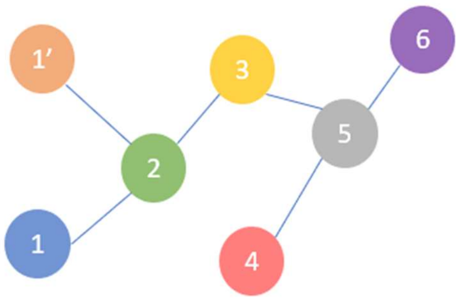
後來詢問教授的建議，發現可以把它以 graph 表示(圖三): 每個 node 代表一個區域，相連的 edge 代表兩個區域有重疊。所以我們要解的是不相連的節點價值總和最大值，而這問題難於「Maximal Independent Set Problem」(最後有證明)，應該稱為「Maximum Weighted Independent Set Problem」，屬於 NP-hard 的問題，故再來要做一些簡化以方便解題。



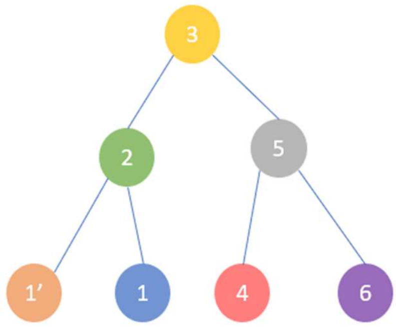
▲ 圖三、Represented by Graph G

簡化: 此問題不包含多個重疊的區域，一個重疊部分最多由兩個區域所貢獻。

=> **graph 沒有 acyclic 的狀況，故我們可以將此 graph 簡化成 tree。**



▲圖四、簡化後的 Graph G



▲圖五、Equivalent Tree of G

當簡化成圖五的時候，比起使用以 Random-priority parallel algorithm 之類 graph 上的近似演算法，可以以更簡單的 Dynamic Programming 來解!

而其中的一個條件: 區域必須不重疊，在 tree 上代表的是如果選了某 node x，就不能選其 parent 與 children，但可以選 grand children。換句話說，類似於紅黑樹中的: No two consecutive red nodes on a simple path。所以從這個「選擇」的想法，我們可以找出一遞迴關係，在此正式定義此問題:

1. Input: A tree T = (V, E) and each node in T has its own weights $w(v) \geq 0$.
2. Goal: Find the maximum weight independent set in T

<Sol>

Let a node x in tree T, by solving this optimization problem, there are two possibilities for node x: either x is chosen or not. Let's discuss these two cases:

Case 1: If x is not chosen as part of the counted weight, its parent and children can be chosen.

Case 2: If x is chosen, we can't choose both its parent and children.

Thus, the recurrence relation can be developed by concluding these two cases:

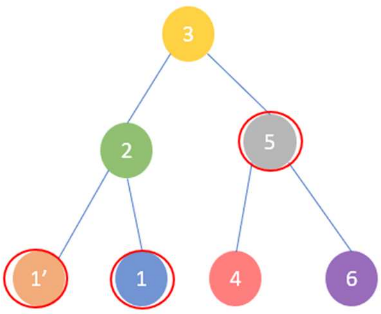
Given T(u) as the subtree of T hanging at node u and Opt(u) represents the max weighted independent set value in T(u).

The recurrence relation is:

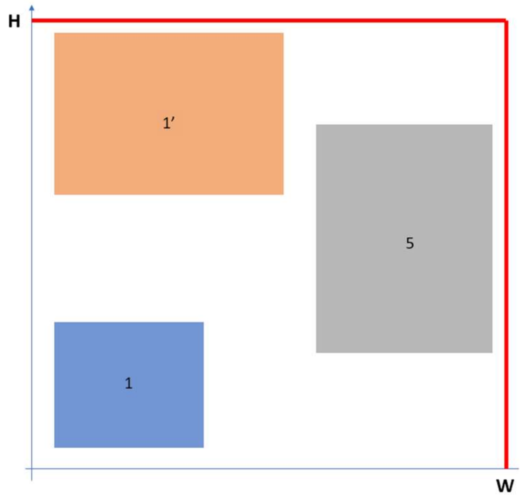
$$\mathrm{Opt(u)} = \max\left(\sum_{\substack{children\,of\,u,\\ as\,v}} Opt(v),\ \ w(u) + \sum_{\substack{grandchildre\\ of\,u,\ as\,x}} Opt(x) \right)$$

And the answer will be Opt(r), where r is the root of T.

By using either memorization or tabulation, this algorithm cost O(n) time. Since each node in T will be calculate only twice (one by its parent, the other by its grandparent). It's worth noting that the recurrence can be solved in the sequence of post-order traversal, which will calculate their children (subtree) first.



▲ 圖六、可能的解



▲ 圖七、對應的結果

鑒於篇幅，就不驗證以上演算法的正確性了，其中一種可能的解如圖六所示，對應的結果在圖七。

最後，來說明一下為何「Maximum Weighted Independent Set Problem ∈ NP-hard」。

根據 J. W. Moon & L. Moser 發表在 Israel Journal of Mathematics volume 中的 On cliques in graphs， "Every graph contains at most $3^{n/3}$ maximal independent sets"，而要計算出 Maximum Weight 在此問題中即是從這些 maximal independent sets 的 weighted sum 中取最大者，代表光是 verifying 就不是 polynomial time 可以完成的，故此問題 ∉ NP。而已知 Maximal Independent Set Problem ∈ NP-hard，而 Maximal Independent Set Problem ≤p Maximum Weighted Independent Set Problem，故此問題∈ NP-hard。

後記：

這次的題目我花了超多時間構思和讀參考資料，雖是辛苦但也獲得不少，下次預計會加入 bin-packing 的概念來解類似的題目，敬請期待。此外，好奇想請問一下，2D Weighted Scheduling Problem 有合適的方法能解嗎? 或是說有近似的演算法可以使用?

B07502173 陳閔鴻

參考資料：

1. NP-Complete Reductions: Clique, Independent Set, Vertex Cover, and Dominating Set
2. https://en.wikipedia.org/wiki/Maximal_independent_set
3. https://courses.grainger.illinois.edu/cs473/sp2011/lectures/09_lec.pdf?fbclid=IwAR0mn6NcpO6Pmzt3joR-GBEscDYleOISVl l3heUcKuJ-TkMiWkikEeU6q8A
4. http://hscc.cs.nthu.edu.tw/~sheujp/public/journal/3.pdf?fbclid=IwAR0Ne3Wektk-KmvgvgZGw00NkI_zFCicjNkPyPnzLEGn4WYGrUkKvQaDRZ4
5. Maximal Independent Set in Graph Theory | Maximal Independent Set Algorithm, Maximum Independent Set