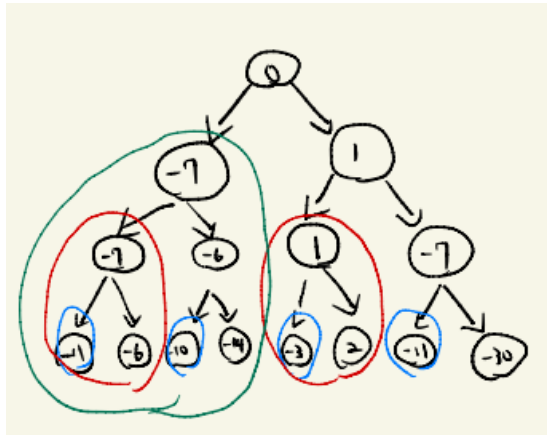


LAB1 Report

0810917 劉峻瑋

一、 如何使用 DP 的方式實現本次題目

若要使用 DP 的方式，就要有 Optimal 的元素和 Subproblem Overlapping 的元素，根據下圖的 tree:

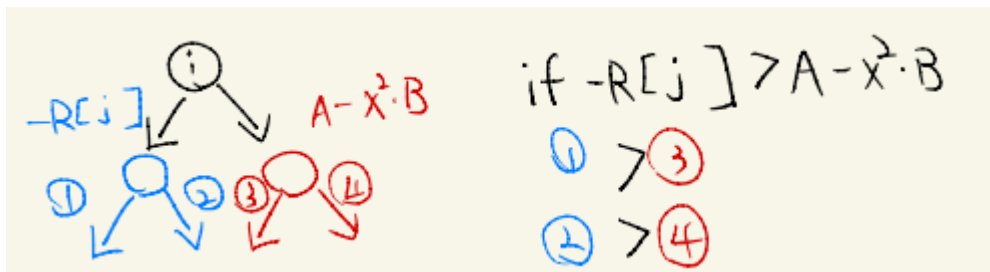


我發現若以日期反推的方法，在休息之後所得到的結果是可以有最佳解的，而此結果在整個 problem 中也會重複使用很多次；因此，我會以 bottom up 的形式從最後一天往前推、儲存每一個休息日之後的最佳解。

二、 如何優化我的程式

若是單純以這種方式建表，會從最後 1 天建表到第 0 天，總共有 n 次，每次要進行的比較會依天數遞減，第 0 天比較 n 次、第 1 天比較 $n-1$ 次、...最後一天比較 0 次。因此時間複雜度計算為 $O(n^2)$ ，在 bigcase 會無法通過。

因此我決定在數字的極限條件下著手，重新審視這個題目，休息和讀書之後的選擇有 4 條路徑，如圖：



會發現若是在當日休息所得到的好處大於讀書所帶來的好處($-R[i] > A - x^2 \cdot B$)，路徑 1:休息後休息>路徑 3:讀書後休息、路徑 2:休息後讀書>路徑 4:讀書後讀書。因此，我們在比較第 i 天要走哪條路時，只要往下找到確定要休息

的第 j 天，從第 j 天，第 $j-1$ 天，第 $j-2$ 天...到第 i 天，就可以找到第 i 天的 optimal 解。

時間複雜度計算:

要找 n 天的 optimal，每天要比較 r 次， r 與 n 無關，因此在 average case 時間複雜度為 $O(n)$ 。

三、 Flow Chart

