|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 2 | T | F | T | F | F | F | F | F | F | F | F | F |
| 3 | T | F | T | T | F | T | F | F | F | F | F | F |
| 7 | T | F | T | T | F | T | F | T | F | T | T | F |
| 8 | T | F | T | T | F | T | F | T | T | T | T | T |
| 10 | T | F | T | T | F | T | F | T | T | T | T | T |

* **子集和問題**
* **子集和問題**是一種NP-complete
* **子集和問題解釋:**

子集和問題,又稱子集合加總問題,是計算複雜度理論和密碼學中一個很重要

的問題。

問題可以描述為:給一個整數集合，判斷是否存在某個非空子集，使子集內的"數字和"為某個特定數值。

譬如，我給個定集合{2,3,7,8,10}，是否存在子集和為11的集合?

答案是，有

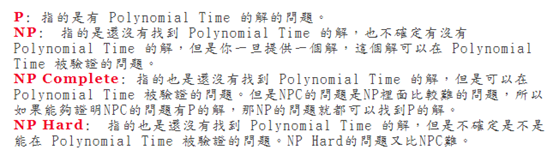
因為子集{3,8}的數字和是11。[1]

* **證明子集和問題是NP-complete:**

1.「問題」本身是一個NP問題

2. NP類中的所有其他問題都可以用多項式時間簡化。

滿足上述兩點可稱為NP-complete



以維基的例子來看

假設給予一組集合{−7,−3,−2,5,8}，問是否有一組子集合相加為0 ?

列舉出所有可能，每個數字都是要選或是不選 ，跟0/1背包問題一樣

-7 + -3 + -2 + 5 + 8

-7 + -3 + -2 + 5

-7 + -3 + -2

-7 + -3

-7

…

-7可選也不選、-3可選也不選、-2可選也不選、5可選也不選、8

可選也不選。

所以有2\*2\*2\*2\*2種可能，2的n次方種可能。 >>2的N次方

而每一項都需要相加確認是不是為”0”。 >>N

所以時間複雜度就是O(N\*(2的N次方))

但假設給另一個定集合{-3, -2, 5}，問使否有子集合相加為0。

那只要-3 + -2 + 5 = 0，時間複雜度O(n)就能驗證答案是正確的

子集和問題符合”一旦有一個解，可以驗證這個解在多項式時間內完成”

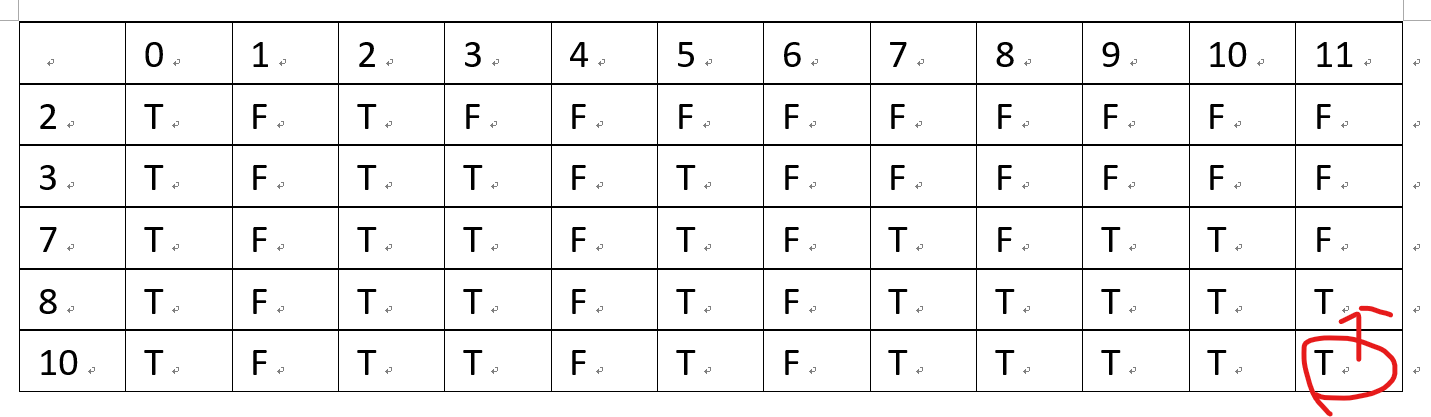
所以子集和問題是NP問題。[2]

而3NF-SAT可以歸約到子集和問題[3]，又3NF-SAT是NP-complete，因此得證子集和問題是NP-complete。

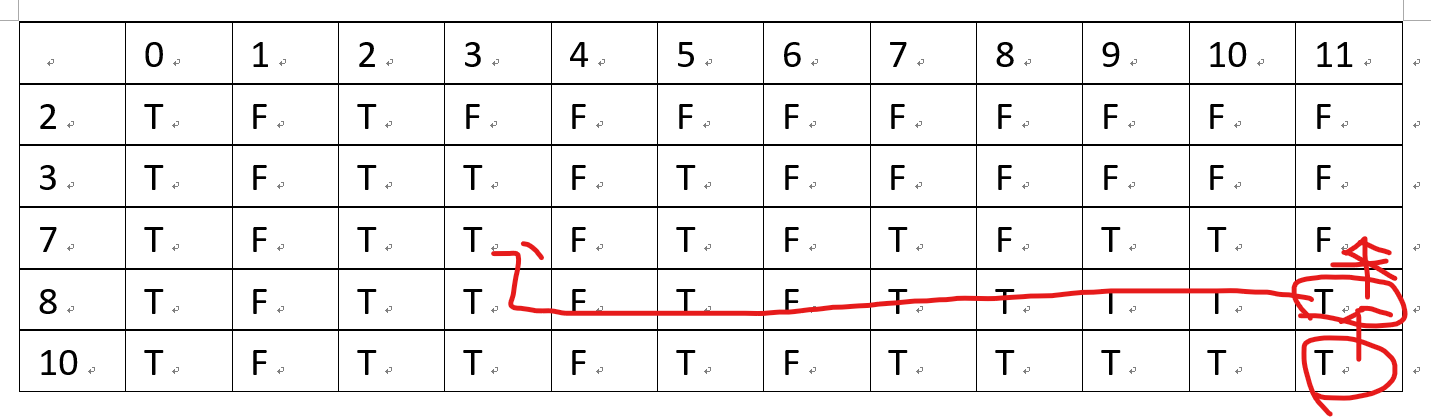
使用動態規劃

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 2 | T | F | T | F | F | F | F | F | F | F | F | F |
| 3 | T | F | T | T | F | T | F | F | F | F | F | F |
| 7 | T | F | T | T | F | T | F | T | F | T | T | F |
| 8 | T | F | T | T | F | T | F | T | T | T | T | T |
| 10 | T | F | T | T | F | T | F | T | T | T | T | T |

這是問題展開後的真值表，橫排是總和，直排是定集合裡的數字。



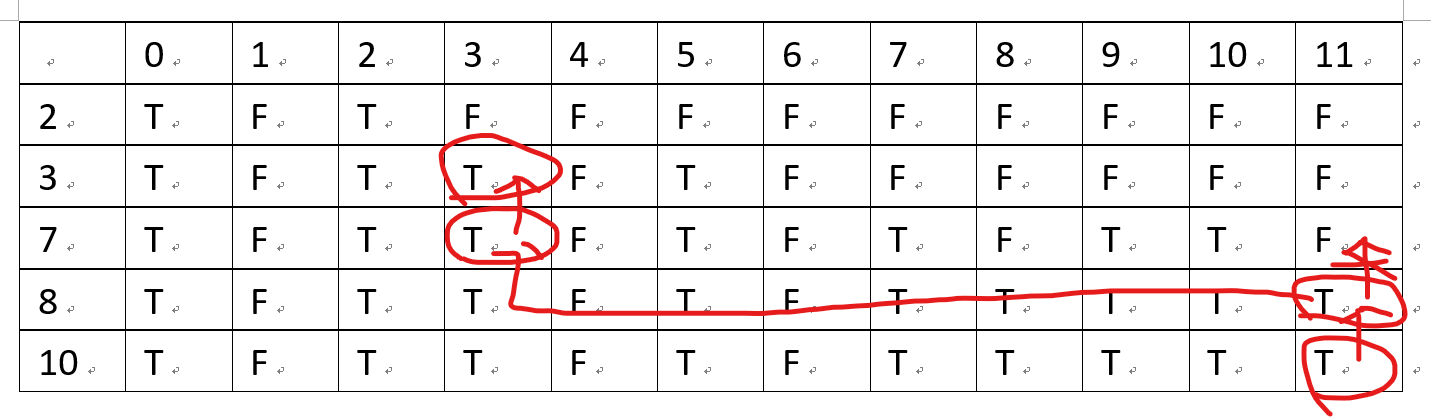
[10,11]的True可以知道是來自於[8,11]的TRUE



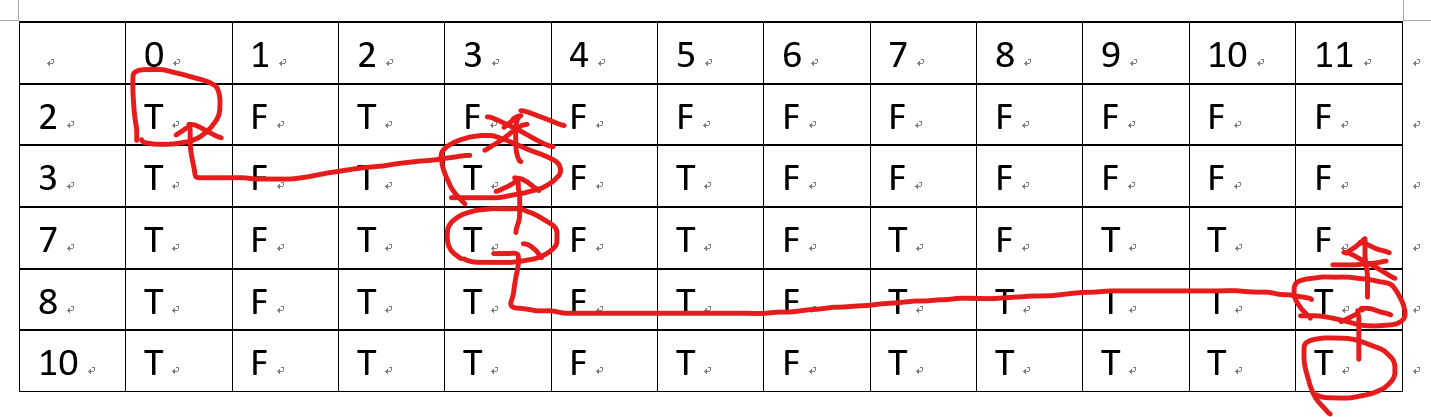
[8,11]的TRUE可以知道不是來自於[7,11]的TRUE，因為[7,11]時是FALSE

所以可以知道答案的子集有8，接下來只要找到是11-8的子集和，也就是3

再來檢查前面，可以發現[7,3]是TRUE



[7,3]的True可以知道是來自於[3,3]的TRUE



[3,3]的TRUE可以知道不是來自於[2,3]的TRUE，因為[2,3]時是FALSE

所以可以知道3要放進答案的子集

之前是{8}

放進3之後，現在子集就是{3,8}，

再來檢查前面的可以發現[3,0]是TRUE，並且能知道是來自於[2,0]的TRUE

流程結束

所以答案子集就是{3,8}。[4]

判斷式

當子集中沒有input[i]

If ( sum < input[i] ) ︿

T[i][j] = T[i-1][j] T[i][j] 換成 T[8-1,11-8]=T[7,3]

else(sum >= input[i]) or ︿

T[i][j] = T[i-1][j] || T[i-1][j-input[i]]

﹀ [5]

[8,11]換掉[10,11]

Input[]={1 3 5 7 9 11}

sum=10

T={1,9} || {3,7}

Input[5]>sum ~意思就是input的11大於10這個sum~

-----------------------------------------------------------------------------------------------------------------

[1] 子集和問題。https://en.wikipedia.org/wiki/Subset\_sum\_problem

[2] Subset Sum Problem。https://ithelp.ithome.com.tw/articles/10240370

[3] Reduction : 3-CNF SAT to Subset Sum。

https://www.youtube.com/watch?v=k8RkYp5KhhU

[4] Subset Sum Problem Dynamic Programming

https://www.youtube.com/watch?v=s6FhG--P7z0

[5] 動態規劃法（三）子集和問題(Subset sum problem)。

https://www.itread01.com/content/1528097890.html