

project3

110502518陳文獻

方法一 標準 DP(button up)

fab62e1982a2

Time Limit Exceeded

Project-3

2000ms

7MB

遞迴式:

```
able[x][y][z] = able[x-candies[i]][y][z] or  
                able[x][y-candies[i]][z] or  
                able[x][y][z-candies[i]]
```

實作上可以壓縮成2維的DP(z可省略)，需要三層迴圈(i, x, y)。

優點:直覺好寫

缺點:陣列空間裡每個點都要走訪，or運算一次拿整個word來運算一個bit，不符時間成本。

時間複雜度估計: $O(\text{糖果數量}^3 * \text{最大糖果重量}^2)$

方法二 bit operation + bitstring (button up)

757d0cf447f2

Accepted

Project-3

339ms

5MB

將原陣列拉平成1D bits string (使用c++ bitset實作)
遞迴式改成如下:

```
able |= (able << (candies[i] * MAX_SUM)) | (able << candies[i]);
```

優點:空間複雜度極小，時間複雜度可接受(解決了or運算的問題)
缺點:每次要shift兩個 10^9 數量級的bits容易stack overflow
時間複雜度同前(但實際在UVA上加速了約20倍)

方法三 top down

16aa0208ac84	Accepted	Project-3	46ms	126MB	糖果差超過100就剪枝
d437c12c3cc0	Accepted	Project-3	463ms	126MB	糖果差超過1000就剪枝

使用遞迴找解，遞迴式、時間複雜度同標準DP。

若不額外加限制，**worst case**會比**button**還差，需要剪枝。

優點:經過多次嘗試可得到極快的結果。

缺點:需要對題目測資探勘，可能因極端測資而超時或**WA**。