## project3

110502518陳文獻

## 方法一標準 DP(button up)

fab62e1982a2

Time Limit Exceeded

Project-3

2000ms

7MB

遞迴式:

```
\label{eq:able_x_gamma} \begin{split} \text{able}[x][y][z] &= \text{able}[x\text{-candies}[i]][y][z] \text{ or} \\ &= \text{able}[x][y\text{-candies}[i]][z] \text{ or} \\ &= \text{able}[x][y][z\text{-candies}[i]] \end{split}
```

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(糖果數量^3\*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:

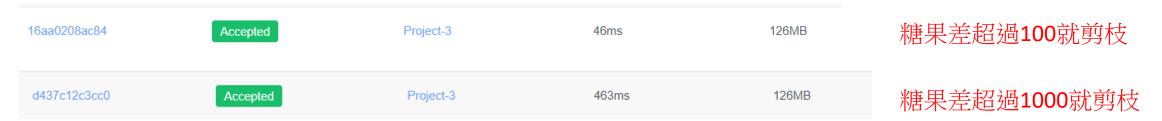
```
able |= (able << (candies[i] * MAX_SUM)) | (able << candies[i]);
```

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



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。