Problem 4 Digit dynamic programming

(1)

dp[n][k]表示長度為 n 開頭為 k 的合法數字數量。

假設其中的合法數字為$a_1 a_2 a_3 \ldots a_{n-2} a_{n-1} a_n$，其中$a_1 = k$

$a_1 a_2 a_3 \ldots a_{n-2} a_{n-1} a_n$的總數是

$a_1 a_2 a_3 \ldots a_{n-2} a_{n-1} 0, a_1 a_2 a_3 \ldots a_{n-2} a_{n-1} 1, \ldots, a_1 a_2 a_3 \ldots a_{n-2} a_{n-1} 9$(也就是 $10 * \mathrm{dp}[\mathrm{n}-1][\mathrm{k}]$)扣掉$a_1 a_2 a_3 \ldots a_{n-2} 38$, $a_1 a_2 a_3 \ldots a_{n-2} 87$(也就是 2*dp[n-2][k])

dp[n][k]=10*dp[n-1][k]-2*dp[n-2][k]

(2)

我認為 dp[n][0]沒有必要。因為 dp[n][0]相當於長度為 n，開頭為 0 的合法數字總數。這部分可以存在 dp[1][0~9]，不用存那麼多組。

(3)

不包含 n，所以 digit 的長度最高到 logn。

因爲 dp table 已經做好，所以所求為

$\sum_{i=1}^{logn} \sum_{j=1}^{9} dp[i][j] + 1$，後面的+1 是算 0 為 legal number

```
ans=1;
for(int i=1; i<logn+1; i++)
        for(int j=1; j<=9; j++){
        ans+=dp[i][j];
}
```

Time complexity 為 O(logn)*O(1)=O(logn)=O(lgn)