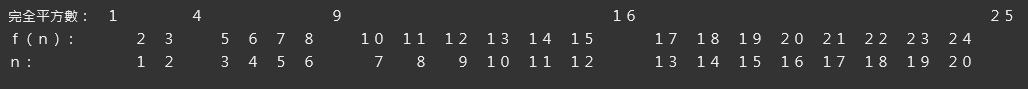
我，想盡辦法，以n表示第n個非完全平方數

作者：1509班18號曹禕中

如標題所示，本題的題目為「**以n表示第n個非完全平方數**」，這題是我在寫程式時遇到的，所以並沒有出處。

### **壹、了解問題**

首先，我先觀察**已知數**「**n**」以及**未知數**「**第n個非完全平方數**」，並將***f(n)***令為「**第n個非完全平方數**」。並將n對*f(n)*做成一張表如下。



### **貳、擬訂計畫**

第一個解法，最容易也是最方便的做法即是**查表**，只要將正整數從一開始排列並剔除完全平方數，且由左向右排序，即可將「**n**」對照到「**第n個非完全平方數**」也就是***f(n)***。但很明顯的，這個方法存在十分明顯的問題，即是不夠靈活，如果今天突然需要第48763個非完全平方數，那豈不是要從1慢慢數起?所以，我就想了第二個解法。

第二個解法，我再次觀察n與*f(n)*，發現之間的**差值洽好等於*f(n)*到1間的完全平方數數量**，並令此差值為x，x = *f(n)* - n，這其實還蠻直觀的，畢竟*f(n)*與n的差本來就是來自經過的完全平方數。這時我們可以列出一些等式了。

### **MSP128910b3h04i51c86ai100001bb0egg8f619e63g參、執行計畫**

*f(n)* - n = x，且n、x為正整數 \*註：[ ]為高斯符號

**MSP273821ie64he733d16810000242fgi39c4d7db0e**再將兩式結合，，當時的我解到這邊就不知怎樣繼續下去了。直到我有一天突發奇想**將高斯符號翻到等號左邊**，結果就變成以下這副模樣。

MSP1074413i0b5he6i2e41db00005d1idd491hacihe1

MSP85151eaa70gicb95ggb600002f3db4f03id0505i

MSP141241heb22c627i3f96200005iiai00507iabci4

MSP81631145383g52d01gc100005fia3bchgad169df

MSP186116f0ief4aga873d000006242888c38igg3da MSP44772343a2ha4e51bdif0000501g45g520b85703

此時負號不會發生，**因為x、n為正整數**，所以不可能是負的。

再將兩式結合如下。

MSP39991fai1c511g6c064d000013247ffg5ided8ee

因為**x為正整數**，且此範圍大於1，可以確認**當n以任意正整數代入時**，**x一定有正整數解**。

此時

MSP160814g8ai39ia78cf41000027eb04307d762799

所以我們現在能用n表示f(n)了對吧?也就是

MSP665915hefe724b5c897f0000330be032g4c2hgf8 或 MSP39221658diah9380g6ee00005e32g880f916065e

咦？**n以2帶入竟然有兩個正整數解**？

我那時非常沮喪，因為一直搞不清楚為何一個n會有多個解，直到兩三天後我終於發現這條列式有問題。

**MSP273821ie64he733d16810000242fgi39c4d7db0e**

這裡的x + n有可能是完全平方數，所以**不能用高斯符號**，不然的話會，應該改成──

MSP513315ibib509a0ii7gb00004369h93id10i5e12x = **小於**的**最大整數**，也就是──

MSP718910b3faghh7i62c3700002i4fbab8gde3bb61

所以以下所有的算式就是將原本的**大於等於**改成**大於**。

IMG_256

IMG_256

IMG_256

IMG_256

IMG_256

IMG_256

這才是正確的計算過程，根據此不等式，*f(n)*的表示方法為──

MSP24391d43e604eb22611900004hd72646c6ad2747 或MSP102123fh3e724f6i2igb00003588ag69452342b1

### **肆、驗算**

最後就讓我用程式來驗證這個公式的正確性吧！

考量其可行性且要方便驗證，故只計算第1到20個非完全平方數。

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

for(float n=1;n<=20;n++){

float fn = floor(pow((n+(0.25)),0.5)+n+(0.5));

if (fn == (pow((n+(0.25)),0.5)+n+(0.5)))

fn--;

cout<<n<<"："<<fn<<endl;

}

}

程式執行結果在下頁

*f(1)*到*f(20)*的數列k如下

k：2 3 5 6 7 8 10 11 12 13 14 15 17 18 19 20 21 22 23 24



結果無誤，公式是正確的，這題也宣告結束，但我的心中還是有一個疑問。

### 伍、回顧

最後的不等式範圍大於等於1，也就是有可能「小於IMG_256的最大整數」不等於「大於IMG_256的最小整數」，雖然目前並沒有發現不等於的情況(我用程式從1檢查到20000000了)，但有可能是我還沒發現而已。第n個非完全平方數應該只有一個，該怎樣證明這個範圍只有一個整數呢？我也不知道，而且也有點超出這題範圍了。這個問題可以請大家回去想想看，如果想到的話也可以和大家分享。

所以這題答案是， MSP24391d43e604eb22611900004hd72646c6ad2747或 MSP102123fh3e724f6i2igb00003588ag69452342b1