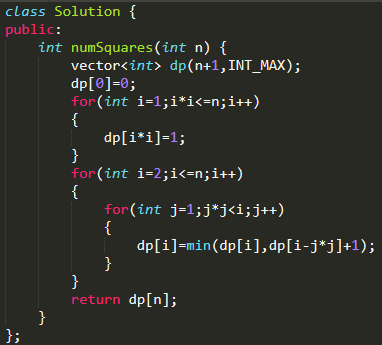
首先分析一下简单版本的代码：



思路相当简单，目的就是将能组成一个数字n的所有加和情况进行遍历。找出其中最小的，算是一种暴力解法。



第二种方法就巧妙了非常多。

第一个巧妙之处在于这个代码使用了Static静态变量来保存数据，因此如果下次查询时，调用的数据已经有了，就不需要再次执行。因此大大提高了代码的执行时间。

第二个最大的巧妙之处在于(也是比较难理解的一个地方):

for (int i = 1; i\*i <= m; i++)

{

cntSquares = min(cntSquares, cntPerfectSquares[m - i\*i] + 1);

}

巧妙和难以理解的地方就在于cntPerfectSquares[m - i\*i] + 1，因为不好理解到底为什么会+1，其实这个地方m-i\*i的意义在于，相当于加上了一个cntPerfectSquares[i\*i]这个数字。这也是加一的原因所在。