题目描述：

Write a program to find the n-th ugly number.

Ugly numbers are**positive numbers** whose prime factors only include 2, 3, 5.

**Example:**

**Input:** n = 10

**Output:** 12

**Explanation:** 1, 2, 3, 4, 5, 6, 8, 9, 10, 12 is the sequence of the first 10 ugly numbers.

**Note:**

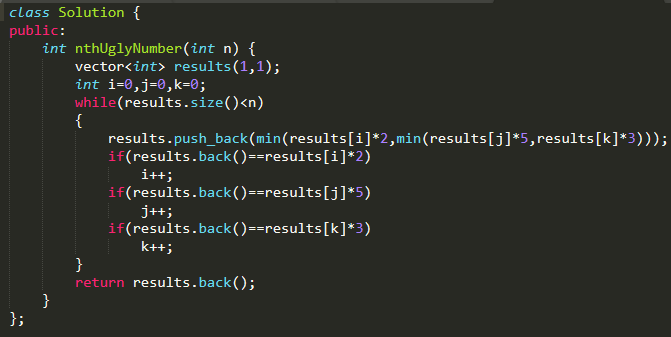
1. 1 is typically treated as an ugly number.
2. n **does not exceed 1690**.
3. **首先要确定好思路**

观察这个题目，其实不难发现，数列中的任何一个数，一定是2 || 3 || 5的倍数。整体的序列排列起来看，一定是这三个数倍数的交叉组合。并且每个数的倍数单独拿出来的话可以发现，一定是有序的。例如，2的0次方，2的1次方，2的2次方，诸如此类。

那么具体的排序就可以得出来。每个位置都是选择三个倍数中最小的。选择好之后只需要将本数的计数数字加一即可。

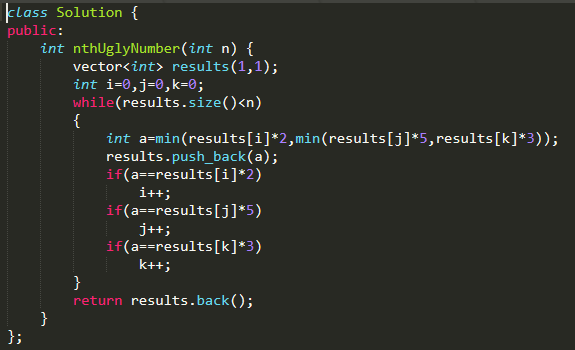
1. **确定好思路以后，如何优化**

**第一个思路：**



可以看到这里面是有冗余计算的，对于vector而言，找result.back()元素其实相当耗费时间。因为vector只能遍历去找这个元素。所以如果在循环中不停的区找result.back()元素，事实上就是一直在做O(n)的遍历，这也就意味着，事实上这个程序的执行时间复杂度为O(n2).

**第二个思路：**



可以看到，在这种思路中，实际上是去掉了while循环里面的遍历部分，提前把值存储了起来。因此时间成功达到了O(n)。

**第三个思路：**

我认为这个思路之所以能够把时间提高到0ms的关键之处在于，线上编译器，也是能够存储Static变量的，并且下次还能直接调用。

