```swift
import Foundation
/// **剑指 Offer 49. 丑数**
/// 我们把只包含质因子 2、3 和 5 的数称作丑数（Ugly Number）。
/// 求按从小到大的顺序的第 n 个丑数。
///
/// 示例:
/// 输入: n = 10
/// 输出: 12
/// 解释: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12 是前 10 个丑数。
///
/// **说明:**
/// - 1 是丑数。
/// - n 不超过1690。
class Solution {
    func nthUglyNumber(_ n: Int) -> Int {
        /// 初始化一个列表用于存储 n 个丑数
        var uglyNums: [Int] = Array(repeating: 1, count: n)
        /// 已知丑数内任意三个丑数的索引
        var (a, b, c) = (0, 0, 0)
        /// 需要注意索引从 1 开始因为第一个数是已知的不需要进行修改
        for i in 1..<n {
            /// 计算三个新的丑数作为候选
            let (newUglyNum1,
                 newUglyNum2,
                 newUglyNum3) = (uglyNums[a] * 2,
                                 uglyNums[b] * 3,
                                 uglyNums[c] * 5)
            /// 当前位置的丑数为候选列表中最小的那一个数
            uglyNums[i] = min(newUglyNum1, newUglyNum2, newUglyNum3)
            /// 如果当前的数uglyNums[i]是由前面某一个索引 index(index = a, b, c) 计算而来
            /// 为了防止重复计算需要将该索引加上1
            if uglyNums[i] == newUglyNum1 { a += 1 }
            if uglyNums[i] == newUglyNum2 { b += 1 }
            if uglyNums[i] == newUglyNum3 { c += 1 }
        }
        return uglyNums[n-1]
    }
}

let solution = Solution()
print(solution.nthUglyNumber(10))
```