https://leetcode-cn.com/problems/chou-shu-lcof/

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. `1` 是丑数。
2. `n` **不超过**1690。

丑数：只包含因子2，3，5的数称作丑数

一个丑数 乘 2，3，5 之后，一定还是一个丑数。所以就会得到丑数（注意7不是丑数）分别乘2，3，5的序列

```
nums2 = {1*2, 2*2, 3*2, 4*2, 5*2, 6*2, 8*2...}
nums3 = {1*3, 2*3, 3*3, 4*3, 5*3, 6*3, 8*3...}
nums5 = {1*5, 2*5, 3*5, 4*5, 5*5, 6*5, 8*5...}
```

最终的丑数就是将 nums2，nums3，nums5合并的序列

dp是丑数的序序列，i代表丑数的下标，p2，p3，p5分别对应nums2，nums3，nums5的对应下标，

1

P2=0, p3=0, p5 = 0

n2= dp[0] * 2

n3= dp[0] * 3

n5= dp[0] * 5

Dp [1] = mins(n2, n3, n5) = 2

Dp[1] == n2 : p2++ = 1

i	**0**	**1**								
dp	1	2								
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

i	0	1	2							
dp	1	2	3							
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

i	0	1	2	3						
dp	1	2	3	4						
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

i	0	1	2	3	4					
dp	1	2	3	4	5					
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

i	0	1	2	3	4	5				
dp	1	2	3	4	5	6				
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

i	0	1	2	3	4	5	6			
dp	1	2	3	4	5	6	8			
		0	1	2	3	4	5	6	7	8
nums2		1 * 2	2 * 2	3 * 2	4 * 2	5 * 2	6 * 2	8 * 2	9 * 2	10 * 2
nums3		1 * 3	2 * 3	3 * 3	4 * 3	5 * 3	6 * 3	8 * 3	9 * 3	10 * 3
nums5		1 * 5	2 * 5	3 * 5	4 * 5	5 * 5	6 * 5	8 * 5	9 * 5	10 * 5

```
def nthUglyNumber(self, n):
        """
        :type n: int
        :rtype: int
        """
        dp,p2,p3,p5 = [1] * n, 0, 0, 0
        for i in range(1,n):
            n2, n3, n5 = dp[p2]*2 , dp[p3] *3, dp[p5] * 5
            dp[i] = min(n2, n3, n5)
             if dp[i] == n2 : p2 += 1
             if dp[i] == n3 : p3 += 1
             if dp[i] == n5 : p5 += 1
         return dp[-1]
```

动态规划

```
class Ugly:
    def __init__(self):
        self.nums = nums = [1, ]
        i2 = i3 = i5 = 0

        for i in range(1, 1690):
            ugly = min(nums[i2] * 2, nums[i3] * 3, nums[i5] * 5)
            nums.append(ugly)

             if ugly == nums[i2] * 2:
                 i2 += 1
             if ugly == nums[i3] * 3:
                 i3 += 1
             if ugly == nums[i5] * 5:
                 i5 += 1

class Solution:
    u = Ugly()
    def nthUglyNumber(self, n):
        return self.u.nums[n - 1]
```

时间复杂度：O（1）时间检索答案和约 1690 * 5 = 8450次预计算操作

空间复杂的：常数空间用保存 1690 个丑数。

方法二：

预计算 1690 个丑数：

初始化预计算用到的数组 nums，堆 heap 和哈希表 seen 跟踪在堆中出现过的元素，避免重复。

循环计算丑数，每个步骤：

弹出堆中最小的数字 k 并添加到数组 nums 中。

若 2k，3k，5k 不存在在哈希表中，则将其添加到堆中并更新哈希表。

返回在数组中预先计算好的丑数。

```python
from heapq import heappop, heappush
class Ugly:
    def __init__(self):
        seen = {1, }
        self.nums = nums = []
        heap = []
        heappush(heap, 1)

        for _ in range(1690):
            curr_ugly = heappop(heap)
            nums.append(curr_ugly)
            for i in [2, 3, 5]:
                new_ugly = curr_ugly * i
                if new_ugly not in seen:
                    seen.add(new_ugly)
                    heappush(heap, new_ugly)

class Solution:
    u = Ugly()
    def nthUglyNumber(self, n):
        return self.u.nums[n - 1]
```

作者：LeetCode

链接：https://leetcode-cn.com/problems/ugly-number-ii/solution/chou-shu-ii-by-leetcode/

来源：力扣（LeetCode）

这种堆的解法是：

每次从堆里弹出最小值，然后再把 k* 2， k*3，k*5 入堆

弹出10次，就是第10丑数，

1 入堆， res = 1

res =heapq.heappop(heap) = 1. 最小值

a,b.c = res * 2, res * 3, res * 5 = 2,3,5

2 2,3,5,入堆

res =heapq.heappop(heap) = 2 最小值

3 4,6,10 入堆，此时堆里为3，5，4，6，10
res =heapq.heappop(heap) = 3 最小值
6，9，15 入堆，此时堆里5，4，6，10，6，9，15

```python
class Solution:
    def nthUglyNumber(self, n: int) -> int:
        import heapq
        heap = [1]
        heapq.heapify(heap)
        res = 0
        for _ in range(n):
            res = heapq.heappop(heap)
            while heap and res == heap[0]:
                res = heapq.heappop(heap)
            a, b, c = res * 2, res * 3, res * 5
            for t in [a, b, c]:
                heapq.heappush(heap, t)
        return res

作者：powcai
链接：https://leetcode-cn.com/problems/ugly-number-ii/solution/dui-
来源：力扣（LeetCode）
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。
```

263

编写一个程序判断给定的数是否为丑数。

丑数就是只包含质因数 `2, 3, 5` 的**正整数**。

示例 1:

```
输入: 6
输出: true
解释: 6 = 2 × 3
```

示例 2:

```
输入: 8
输出: true
解释: 8 = 2 × 2 × 2
```

示例 3:

```
输入: 14
输出: false
解释: 14 不是丑数，因为它包含了另外一个质因数 7。
```

说明:

1. `1` 是丑数。
2. 输入不会超过 32 位有符号整数的范围: $[-2^{31}, 2^{31} - 1]$。

```python
# 可以递归，可以循环，都是不断除 2，3，5三个质因数，不过注意添加，num =1和num
class Solution(object):
    def isUgly(self, num):
        if num == 1:
            return True
        if num < 1:
            return False
        while num != 1:
            if num % 5 == 0:
                num /= 5
            elif num % 3 == 0:
                num /= 3
            elif num % 2 == 0:
                num /= 2
            else:
                return False
        return True

```