

剑指Offer（三十三）：丑数

🕒 2018年1月8日 10:16:03 💬 2 🌡 3,221 °C 📝 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

把只包含因子2、3和5的数称作**丑数**（Ugly Number）。例如6、8都是**丑数**，但14不是，因为它包含因子7。习惯上我们把1当做是第一个丑数。求按从小到大的顺序的第N个丑数。

1、思路

所谓的一个数m是另一个数n的因子，是指n能被m整除，也就是 $n \% m == 0$ 。根据丑数的定义，丑数只能被2、3和5整除。根据丑数的定义，丑数应该是一个丑数乘以2、3或者5的结果（1除外）。因此我们可以创建一个数组，里面的数字是排好序的丑数，每一个丑数都是前面的丑数乘以2、3或者5得到的。

这个思路的关键问题在于怎样保证数组里面的丑数是排好序的。对乘以2而言，肯定存在某一个丑数T2，排在它之前的每一个丑数乘以2得到的结果都小于已有最大的丑数，在它之后的每一个丑数乘以乘以2得到的结果都会太大。我们只需要记下这个丑数的位置，同时每次生成新的丑数的时候，去更新这T2。对乘以3和5而言，也存在着同样的T3和T5。

C++:

C++

```
1 class Solution {
2 public:
3     int GetUglyNumber_Solution(int index) {
4         if(index < 7){
5             return index;
6         }
7         vector<int> res(index);
8         for(int i = 0; i < 6; i++){
9             res[i] = i + 1;
10        }
11        int t2 = 3, t3 = 2, t5 = 1;
12        for(int i = 6; i < index; i++){
13            res[i] = min(res[t2] * 2, min(res[t3] * 3, res[t5] * 5));
14            while(res[i] >= res[t2] * 2){
15                t2++;
16            }
17            while(res[i] >= res[t3] * 3){
18                t3++;
19            }
20            while(res[i] >= res[t5] * 5){
21                t5++;
22            }
23        }
24        return res[index - 1];
25    }
26};
```

注意：1，2，3，4，5，6都是丑数。所以当index小于7的时候，直接返回index即可。

Python:

Python

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def GetUglyNumber_Solution(self, index):
4         # write code here
5         if index < 7:
6             return index
7         res = [1, 2, 3, 4, 5, 6]
8         t2, t3, t5 = 3, 2, 1
9         for i in range(6, index):
10             res.append(min(res[t2] * 2, min(res[t3] * 3, res[t5] * 5)))
11             while res[t2] * 2 <= res[i]:
12                 t2 += 1
13             while res[t3] * 3 <= res[i]:
14                 t3 += 1
15             while res[t5] * 5 <= res[i]:
16                 t5 += 1
17         return res[index - 1]
```



微信公众号

分享技术，乐享生活：微信公众号搜索
「JackCui-AI」关注一个在互联网摸爬滚
打的潜行者。

因为失去，所以明白。--- 卡卡西《火影忍者》