313. Super Ugly Number

Medium 320 81 Favorite Share

Write a program to find the n^{th} super ugly number.

Super ugly numbers are positive numbers whose all prime factors are in the given prime list `primes` of size `k`.

Example:

```
Input: n = 12, primes = [2,7,13,19]
Output: 32
Explanation: [1,2,4,7,8,13,14,16,19,26,28,32] is the sequence of the first
12
             super ugly numbers given primes = [2,7,13,19] of size 4.
```

Note:

- `1` is a super ugly number for any given `primes`.
- The given numbers in `primes` are in ascending order.
- $0 <$ `k` ≤ 100, $0 <$ `n` $\le 10^6$, $0 <$ `primes[i]` < 1000.
- The n^{th} super ugly number is guaranteed to fit in a 32-bit signed integer.

Accepted 56,641 | Submissions 139,833

Seen this question in a real interview before? Yes No

Contributor

```java
// 可以先看L264，丑数2，思想基本一样
public class L313 {
    public int nthSuperUglyNumber(int n, int[] primes) {
        int [] dp = new int[n];
        //第一个丑数为1
        dp[0] = 1;
        int [] idxPrimes = new int [primes.length];
        int counter = 1;
        while (counter < n) {  //这里的思想和丑数2是一样的
           int min = Integer.MAX_VALUE;
           for(int i = 0; i < primes.length; i ++) {
               int temp = dp[idxPrimes[i]] * primes[i];
               min = min < temp ? min : temp;
           }
           /*
            * 这里因为比如说primes为 2 7 13 19
            * idxPrimes[i]开始全部为0，经过第一轮后，idxPrimes[0] = 1,其余为0，dp[0] = 1
            * 这样新一轮temp下来，依次为4 7 13 19，idxPrimes[0] = 2,其余为0，dp[1] = 2
            * 新一轮temp下来，依次为8 7 13 19，idxPrimes[0] = 3,其余为0，dp[2] = 4
            * 新一轮temp下来，依次为16 7 13 19，idxPrimes[0] = 3,idxPrimes[1] = 7，其余为0，dp[3] = 7
            * 记录idxPrimes[i]的原因就是要获得最小的那个质数，这样下次就是7 * dp[1] = 14了，因为7 * dp[0]已经用过了
            */
           for(int i = 0; i < primes.length; i ++) {
               if(min == dp[idxPrimes[i]] * primes[i])
                   idxPrimes[i] ++;
           }
           dp[counter] = min;
           counter ++;
       }
        return dp[n - 1];
    }
}
```