Problem 313: Super Ugly Number

Problem Information

Difficulty: Medium
Acceptance Rate: 45.74%
Paid Only: No
Tags: Array, Math, Dynamic Programming

Problem Description

A **super ugly number** is a positive integer whose prime factors are in the array `primes`.

Given an integer `n` and an array of integers `primes`, return _the_ `nth` _**super ugly number**_.

The `nth` **super ugly number** is **guaranteed** to fit in a **32-bit** signed integer.

Example 1:

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].

Example 2:

Input: n = 1, primes = [2,3,5] **Output:** 1 **Explanation:** 1 has no prime factors, therefore all of its prime factors are in the array primes = [2,3,5].

Constraints:

* `1 <= n <= 105` * `1 <= primes.length <= 100` * `2 <= primes[i] <= 1000` * `primes[i]` is **guaranteed** to be a prime number. * All the values of `primes` are **unique** and sorted in **ascending order**.

Code Snippets

C++:

```
class Solution {
public:
int nthSuperUglyNumber(int n, vector<int>& primes) {

}
};
```

Java:

```
class Solution {
public int nthSuperUglyNumber(int n, int[] primes) {

}
}
```

Python3:

```
class Solution:
def nthSuperUglyNumber(self, n: int, primes: List[int]) -> int:
```