

# Analysis: Consecutive Primes

## Test Set 1

Say  $P(n)$  denotes the number of primes up to  $n$ , then the number of possible products of consecutive primes will be  $P(n) - 1$ . For this test set, we can find all the primes up to  $n = 2021$  and generate the list of products of consecutive primes. Each query can then be answered by searching for the largest product which is still smaller than or equal to the given query.

*Complexity analysis:*

1. To generate primes, we can test all  $i$  up to  $n$  for primality by checking if any integer from 2 to  $i - 1$  divides  $i$ . This takes  $\sum_{i=0}^{n-2} i = O(n^2)$
2. Generating the list of products of consecutive primes from the prime list takes  $O(n)$  (a tighter bound will be  $O(P(n))$  but we do not require it for this case)
3. To answer all the queries, the complexity is  $O(T \times n)$

So, the total complexity is  $O(n^2) + O(T \times n)$  which is well under the required time limit for  $n = 2021$ .

## Test Set 2

We do not need to generate all the primes up to  $n$ , but only until we find a prime which is greater than  $\sqrt{n}$ . The product of consecutive primes larger than  $\sqrt{n}$  will be greater than  $n$ . For this test set,  $n$  can be up to  $10^9$ , so we can generate primes up to  $10^5$  (rounding off  $10^{4.5}$  to cover the one up case as well) using [Sieve of Eratosthenes](#).

*Complexity analysis:*  $O(\sqrt{n} \times \log(\log(\sqrt{n})))$  (for sieve) +  $O(T \times P(\sqrt{n}))$  (for queries) which is well under the required time limit for  $n = 10^9$ .

## Test Set 3

Our previous strategy of generating primes up to  $\sqrt{n}$  will not work in this case as  $\sqrt{n}$  can be as large as  $10^9$ .

For any input  $n$ , there can be only two scenarios:

- The solution is the product of the largest prime smaller than or equal to  $\sqrt{n}$  and the smallest prime greater than  $\sqrt{n}$ . Product of consecutive primes larger than that will be larger than  $n$ .
- The solution is the product of two largest primes smaller than or equal to  $\sqrt{n}$ . This will be less than  $n$  and any product of consecutive primes less than that will be even smaller.

So, we need primes only in the vicinity of  $\sqrt{n}$ , specifically in the worst case, 2 primes smaller than or equal to  $\sqrt{n}$  and one prime larger than  $\sqrt{n}$ .

Now, how far out can the closest prime be from any given  $\sqrt{n}$ ? This series of numbers is known as [maximal prime gaps](#) and for  $\sqrt{n} = 10^9$  (for maximum possible  $n = 10^{18}$ ), this number is 282 (called *max\_gap* from now on). So, we can iterate backwards from  $\sqrt{n}$  until we find 2 primes

and then iterate ahead from  $\sqrt[4]{n}$  for 1 more prime. In the worst case we would need to check integers in range  $[\sqrt[4]{n} - \text{max\_gap} \times 2, \sqrt[4]{n} + \text{max\_gap}]$ .

*Complexity analysis:*  $O(\sqrt[4]{n} \times 3 \times \text{max\_gap} \times T)$   
(primality check over  $3 \times \text{max\_gap}$  possible integers for  $T$  test cases)

*Note:* The above complexity is not a tight bound. It's hard to hit the worst case scenario and even then most of the primality checks will terminate earlier (using just the first 7 primes, we can rule out 662 integers in any  $3 \times 282$  range). A better [primality test](#) can be used to further drive the complexity down. [Segmented sieve](#) can also be used to have deterministic complexity -  $O((\sqrt[4]{n} + 3 \times \text{max\_gap} \times T) \times \log(\log(\sqrt[4]{n})))$ .