Generating Goldbach - Solution

The main challenge of this problem is to generate the necessary primes sufficiently fast. As $n \leq 10^5$, the required primes are also at most 10^5. These primes can be quickly generated by using the sieve of Eratosthenes; create a boolean array with 10^5 entries, initially all set to true. These indicate whether the corresponding index is prime. Set 1 to not be a prime and next we iterate over the elements of the array. If an element is currently set to true, it is a prime and we set all multiples of the index to false. If we traverse the array from small to big, this will correctly generate all primes up to 10^5. It should be noted that these primes can also be generated inefficiently and stored in a hard-coded array. However care should be taken to ensure the submission file size limit is not exceeded.

Given n, we now enumerate all primes $p \leq 10^5$ and check if $n - p$ is a prime as well. If so, return p and $n - p$. As the Goldbach conjecture holds up to 10^5, this will always return a valid answer. Ensure to only give a single answer, even if multiple are possible. Take care to efficiently check whether $n - p$ is prime, for instance by checking the boolean array. Many different approaches are possible, for instance using the Miller-Rabin primality test and enumerating all odd numbers and 2 instead of all primes p. The very basic approach of enumerating all pairs of primes and checking if they sum to n will however be too slow. As a final note, ensure taking into account $p = 2$ as well! As $n - 2$ is always even, $p = 2$ cannot be used for $n > 4$, but for $n = 4$ it is in fact necessary.