

Explanation

1. Bug Fix:

- **Original Issue:** The code used a list as a default argument, which caused unexpected behavior(because list persist across multiple function calls) when calling the function multiple times.
- **Fix:** Changed the default argument to None and initialized the list inside the function.

2. Optimizations:

- Removed all factors of 2 at the start. This helps avoid unnecessary checks for even numbers later.
- **Check only odd numbers:** After handling 2, only check odd numbers (skip all even numbers). This cuts the number of checks by half.
- **Stop early if the remaining n is prime:** After the loop, if n is still greater than 1, it must be prime, so add it directly to the list of factors.

Time Complexity

- **Original Version:** it checks all numbers up to the square root of n.
- **Optimized Version:** Still $O(\sqrt{n})$ but with fewer steps:
 - It skips even numbers and stops early if n is prime.

Performance Comparison

- **Original Version:** ~0.0020 seconds.
- **Optimized Version:** ~0.0015 seconds.

```
PS C:\Users\softe\OneDrive\Documents\python_script_course> python optimized_prime_factor.py
([], 0.0)
([2], 0.0)
([2, 5, 5], 0.0)
([71, 887, 158788129], 0.0015163421630859375)
PS C:\Users\softe\OneDrive\Documents\python_script_course> python prime_1.py
PS C:\Users\softe\OneDrive\Documents\python_script_course> python prime_1.py
([], 0.0)
([2], 0.0)
([2, 5, 5], 0.0)
([71, 887, 158788129], 0.002026081085205078)
```

- PS C:\Users\softe\OneDrive\Documents\python_script_course> █