① Current Optimized code:

```
- def isPrime(x):
-     ...
-     while i <= root_x:
-         if x % i == 0:
-             return False
+ def generatePrimes(n):
+     sieve = []
+     ...
+     while p * p <= n:
+         ...
+         for i in range(p * p, n + 1, p):
+             sieve[i] = False
```

Incorrect initailization

Low input similarity

② The Similar Pattern:

```
- for VAR in range(NUM, int(VAR**NUM)+NUM):
-     ...
+ def prime_table(NUM):
+     ...
+     VAR = [BOOL] * (NUM + 1)
+     for NUM in range(NUM, int(VAR**NUM)+NUM):
```

③ Refined code:

```
def generatePrimes(n):
    sieve = [true] * (n+1)
    ...
    while p * p <= n:
        ...
        for i in range(p * p, n + 1, p):
            sieve[i] = False
```