Methods for generating a list of primes

Method 1:

a.) First, check if n=2. If it is, n is prime. Otherwise, proceed to step b.

b.) Check to see if each integer k is a divisor of n where 2 < k < n. If none of the values of k are divisors of n, then n is prime

Method 2:

a.) First, check if n=2. If it is, n is prime. Otherwise, proceed to step b.

b.) Check to see if each integer k is a divisor of n where $2 < k \leq \sqrt{n}$. If none of the values of k are divisors of n, then n is prime

Method 3:

a.) First, check if n=2. If it is, n is prime. Otherwise, proceed to step b.

b.) Check if n is divisible by 2. If so, n is not prime. Otherwise, proceed to step c.

c.) Check to see if each odd integer k is a divisor of n where $2 < k \leq \sqrt{n}$. If none of the values of k are divisors of n, then n is prime.

Method 4:

a.) First, check if n=2. If it is, n is prime. Otherwise, proceed to step b.

b.) Check to see if each prime integer k is a divisor of n where $2 < k \leq \sqrt{n}$. If none of the values of k are divisors of n, then n is prime.

Exercise:

Sketch out the code for each of these methods to determine the number of times a specific operation occurs. Use the table below to compare the number of types of operations (compares, division checks, assignments, counter (k) modifications, and jumps) of the different methods.

	compares	division checks	assignments	counter modifications	jumps	total
Method 1						
Method 2						
Method 3						
Method 4						

Using your answers in the table above, identify the rate at which the complexity of each method grows with different values of n.