

```
public static List<Integer> sieveOfEratosthenes(int n) {  
    boolean[] prime = new boolean[n + 1];  
    for (int i = 0; i <= n; i++)  
        prime[i] = true;  
  
    for (int p = 2; p * p <= n; p++) {  
        if (prime[p]) {  
            for (int i = p * p; i <= n; i += p)  
                prime[i] = false;  
        }  
    }  
  
    List<Integer> primes = new ArrayList<>();  
    for (int i = 2; i <= n; i++) {  
        if (prime[i]) {  
            primes.add(i);  
        }  
    }  
    return primes;  
}
```

The diagram illustrates the control flow of the sieveOfEratosthenes method. It features several nodes (circles) and edges (lines) connecting them. The nodes are colored in shades of orange and yellow. The edges represent the flow of execution between different parts of the code, such as loops and conditional statements. For example, a node is connected to the start of the first loop, and another node is connected to the start of the second loop. The diagram highlights the flow from the initialization of the prime array to the final return statement.