

Primes

This algorithm runs by iterating from 2 through the maximum integer, comparing each current iteration to the list of stored primes. If none of the primes can cleanly divide ($x\%y=0$) the number in question, it is added to the list of primes. At the end of the iteration the primes are returned. Also the code checks if the number is less than one and returns nothing (this was excluded from the pseudocode for concision)

Pseudocode:

```
int n = stdin int[] primes = new int[]
for (int i=2; i<n; i++){
    bool isPrime = true
    for(int x=0; x<primes.length; x++){
        if( i%primes[x]==0 ){
            isPrime= false
            break
        }
    }
    if(isPrime){
        primes.add(i)
    }
}
return primes;
```

Because the base factors of any non-prime number will be prime, by dividing a given number by all previous primes there is no possibility of marking wrong. If no prime can divide the number in question cleanly, this number is also prime - it can only be divided by itself and 1. Otherwise it can be divided by another number, therefore not prime.

I believe because this function iterates through every recorded prime this algorithm operates in **$O(n!)$**