

CSCI-261-05 Homework 00 Primes

Kieran Stewart

Algorithm Description

My algorithm goes through each number up until the number n . For each number it checks if it is a multiple of any of the primes that have been discovered so far. If it is not, then it must be prime and is added to the array.

Psudo Code

```
getPrimes (n):  
    if n is less than 2 then return an empty array  
  
    create an empty array to hold the primes  
    let i = 2  
    for each number i from 2 to n (inclusive):  
        check if i is divisible by the numbers in the primes array.  
        if it isn't, add it to the primes array.  
    return the primes array
```

Proof

We go through each number from 2 to n , for each number there are two cases. If the number is not prime, since we know that every non prime number is the product of two prime numbers, it will be divisible by one of the numbers in primes. If the number is prime, then it will have no factors, and it will not be divisible by any of the numbers in primes.

Running Estimate

$O(n^2)$

The program runs in $O(n^2)$ time, which we can tell because it runs through each number from 1 to n and for each of those it runs an operation from 2 to the length of primes. Primes can be as long as n (as in the case of 2), so we get $n \times n$ or n^2