ISPRIME(inputNumber, possibleFactor) (n-1)

‘’’ function takes in an integer larger than one (as otherwise it cannot be a prime number so there is no point checking). It also takes in another integer which is less than the first integer. Function continually checks if inputNumber divided by any number up until 1 can be divided without a remainder. If it can be, then it must not be a prime number and thus returns False. If it cannot be it calls the function again with the possibleFunction variables value being one less than last time. If the possibleFactor value becomes equal to one. Then the input number is prime and True is returned.’’’

//if there has not been a number which it can be divided by without a remainder by before one then it is a prime.

If possible factor = 1 (n-1)

Return True (1)

//if current number cannot be divided by possibleFactor without a remainder then call //function again with the value of possibleFactor being one less.

Else If inputNumber mod possibleFactor ≠ 0 (n-2)

Return ISPRIME(inputNumber, possibleFactor-1) (n-2)

Else (1)

//if the number can be divided without a remainder then it is not a prime number.

Return False (1)

Run time: (4n-6) + 3

Big O: O(n)