Problem 1

Problem 3

```
public boolean isPrime(int n){
   if(n <= 1) return false;
   if(n == 2) return true;
   if(n > 2 &&  n%2 == 0) return false;
   int maxDivisor = (int)Math.sqrt(n);
   for(int divisor = 3; divisor < maxDivisor+1; divisor+=2){
           if(n%divisor== 0) return false;
   }
   return true;
}
```

The asymptotic running time is O(n^1/2). This is because first off it only checks for odd numbers and check only those less than the square root of the number