**Modification for increasing efficiency of Algorithm to find prime Number:**

1. Add 2 in the sieve and increment the counter of the loop by factor of 2.
2. Use i \* i < n rather than using i < sqrt(n). as sqrt used more operation than multiplication.
3. Asd

Algo For sieve:

public boolean[] sieve(int n)

{

boolean[] prime=new boolean[n+1];

Arrays.fill(prime,true);

prime[0]=false;

prime[1]=false;

int m=Math.sqrt(n);

for (int i=2; i<=m; i++)

if (prime[i])

for (int k=i\*i; k<=n; k+=i)

prime[k]=false;

return prime;

}

**GCD:**

Euclid Algorithm:

//assume that a and b cannot both be 0

public int GCD(int a, int b)

{

if (b==0) return a;

return GCD(b,a%b);

}

**LCM:**

The formula for LCM is: LCM(a, b) = (a\*b) / GCD(a,b);

Algorithm For LCM:

public int LCM(int a, int b)

{

return b\*a/GCD(a,b);

}