P(x) / 1 -> oo Euclid
(prime numbers are infinite)

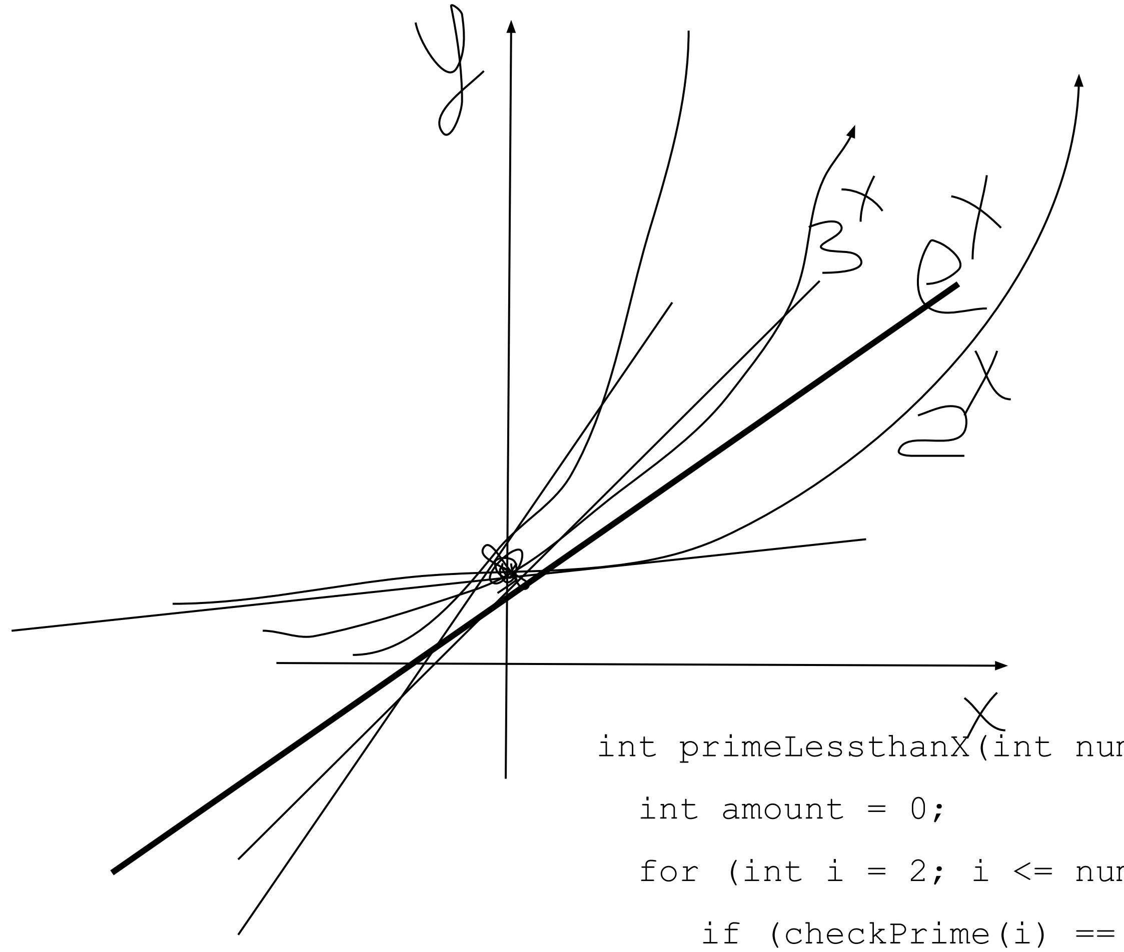
Distribution of primes

P(n) - amount of primes less than n
P(20)=[3,5,7,11,13,17,19]=7

P(x) / x -> 0 x->oo
(XVIII - Leonard Euler)

x=1000000
x/ln(x)=1000000/ln(1000000)

P(x) / x/ln(x) -> 1
(XIX Chebyshev) e=2,71…..



log_a(b)=log_c(b)/log_c(a)
log_e(b)=log_c(b)/log_c(e)

```
int primeLessthanX(int number) {
  int amount = 0;
  for (int i = 2; i <= number; i++) {
     if (checkPrime(i) == 1) {
        amount++;
        //std::cout << i << std::endl;
     }
  }
  std::cout << amount << std::endl;
  return amount;
}

int number = 20000000;

int pX = primeLessthanX(number);
int pY = (double)number / log(number);
std::cout << pY << std::endl;
std::cout << ((double)pY / (double)pX) * 100 << std::endl;
```

Bertrand's postulate (there is at least one prime number between n and 2n)
x/ln(x) ~ P(x)
2x/ln(2x) ~ P(2x)

P(2x)-P(x)=2x/ln(2x) - x/ln(x) =
=x[2/ln(2x) - 1/ln(x)] > x[2/ln(2x) - 1/ln(2x)]=x[1/ln(2x)]=x/ln(2x)>1