Number of Primes in $n^{2/3}$

If $x \in [2, n]$, x – composite

$\Rightarrow$ x has divisor $y \leq n^{1/2}$

Let $k = \lfloor n^{1/2} \rfloor$ (or $k \geq \lfloor n^{1/2} \rfloor$)

Let $P = \{p_1 \dots p_t\}$ be primes in $1 \dots k$

How many primes are in $k+1 \dots n$?

$F_t(n) - 1$, where

$$F_t(n) = |\{x \in [1; n] \mid \forall i \in [t]\ x \not\vdots p_i\}|$$

(subtract 1, because we overcounted $x=1$)

need to compute this

So total count of primes in $[1, n]$ is $t + F_t(n) - 1$.

Let $cnt_n(g) = |\{x \in [1; n] \mid x \vdots g\}| = \lfloor \frac{n}{g} \rfloor$

$$F_t(n) = |\{x \in [1; n] \mid \forall i \in [t]\ x \not\vdots p_i\}|$$
$$= n - cnt_n(p_1) - cnt_n(p_2) - \dots cnt_n(p_t)$$
$$+ cnt_n(p_1 p_2) + cnt_n(p_1 p_3) + \dots + cnt_n(p_{t-1} p_t)$$
$$- cnt_n(p_1 p_2 p_3) - \dots$$

Let $Rec_0(n) = n$

$$Rec_i(n) = Rec_{i-1}(n) - Rec_{i-1}\left(\left\lfloor \frac{n}{p_i} \right\rfloor\right)$$

Claim (By induction)

$$Rec_t(n) = n - cnt_n(P_1) - cnt_n(P_2) - \dots cnt_n(P_t)$$
$$+ cnt_n(P_1 P_2) + cnt_n(P_1 P_3) + \dots + cnt_n(P_{t-1} P_t)$$
$$- cnt_n(P_1 P_2 P_3) - \dots$$

Aka $F_t(n) = Rec_t(n)$

<u>Algorithm</u> (slow)

```
Ans = 0
Rec(t, n, sign):
    if t == 0
        Ans += sign · n
        return

    Rec(t-1, n, sign)
    Rec(t-1, ⌊n/P_t⌋, -sign)
```

Algorithm (fast)

Ans = 0

Rec(t, n, sign):

  if $t == 0$

    Ans += sign $\cdot$ n

    return

  if $n \le Lim$:

    Defer query (t, n, sign)

    return

  Rec(t − 1, n, sign)

  Rec$(t-1, \lfloor \frac{n}{p_t} \rfloor, -sign)$

Deferred queries can be processed in

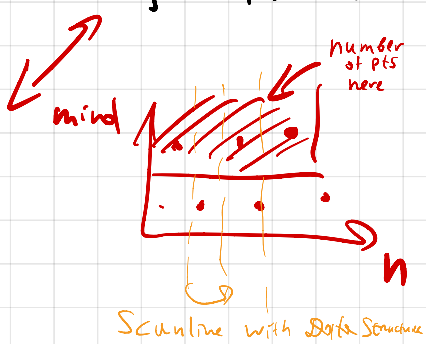
$$O((Lim + CountQueries) \cdot \log t)$$

Process deferred queries $n \uparrow$

[see linear sieve algorithm]

$$(t, n, sign) \rightarrow sign \cdot \left| \{ x \in [1, n] \mid \underbrace{mind[x] > t\text{-th prime}}_{\text{prefix sum in fenwick tree for already added numbers}} \} \right|$$

$\uparrow$ Count among 1..n numbers not div by $p_1 \dots p_t$



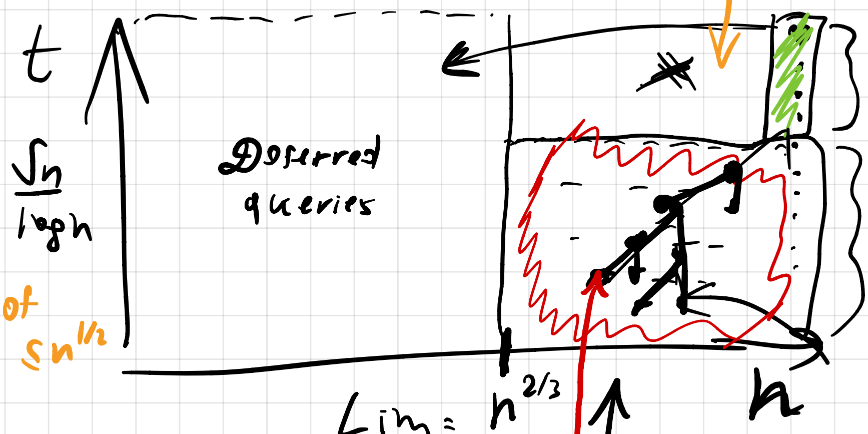
Complexity

every state has this form:

$$\left(t, \left\lfloor \frac{n}{k} \right\rfloor, \text{sign}\right)$$

note: no queries here

Because just one division by p_i makes us go $\leq Lim$



$\frac{\sqrt{n}}{\log n}$ ← Count of primes $\leq n^{1/2}$

$\frac{n^{1/2}}{\log n} - \frac{n^{1/3}}{\log n}$

$\frac{n^{1/3}}{\log n}$ ← also count of primes $\leq \frac{n}{Lim} = n^{1/3}$

only values $\underbrace{\frac{n}{1} \ldots \frac{n}{n/Lim}}_{n^{1/3}} = \frac{n}{n^{1/3}}$

no k repeats in $\lfloor \frac{n}{k} \rfloor$

Because $k = p_1 \cdot p_2 \cdot p_3$ for chosen primes

$$Nodes = O\left(\frac{n^{1/2}}{\log n} - \frac{n^{1/3}}{\log n} + \frac{n}{Lim} \cdot \frac{n/Lim}{\log n}\right) = O\left(\frac{n^2/Lim^2}{\log}\right)$$

too small to care abt

$$\text{Online Time} = \Theta(Nodes)$$

$$\text{Offline Time} = O\big((Lim + \underset{O(Nodes)}{\underset{\shortparallel}{Queries}}) \log \underset{\log n/2}{\underset{\shortparallel}{\sqrt{n}}}\big) =$$

$$= O(Lim \log n + Nodes \log n)$$

$$\text{Total } T = O\left(Lim \log n + \frac{n^2/Lim^2}{\log n} \cdot \log n\right)$$

$$= O(Lim \log n + n^2/Lim^2)$$

$= O(n^{2/3} \log)$ if $Lim = n^{2/3}$

$= O(n^{2/3} \log^{1/3})$ solving $Lim \cdot \log = \frac{n^2}{Lim^2}$

Linear Sieve

- Find primes
- Compute multiplicative functions
- Find min divisor for every $[2; N]$

do this, everything else is possible on top

$n \in [2; N]$

$$n = p_1^{k_1} p_2^{k_2} \dots p_t^{k_t} \longrightarrow \left(p_1, \frac{n}{p_1}\right)$$

min divisor of n

min divisor $\geq p_1$)

```
primes = []
min_divisor_id = [-1 for i=0..N]

For n = 2..N:
    if min_divisor_id[n] == -1:
        primes.append(n)
        min_divisor_id[n] = len(primes) - 1
    for (j = 0;
         j ≤ min_divisor_id[n] && n·primes[j] ≤ N;
         j++)
        min_divisor_id[n·primes[j]] = j
```

discover each number exactly once.

$O(n)$

Sum of φ

"Magic trick"

Theorem $Id = 1 * \varphi$

Definitions:

$$Id(n) = n$$

$$1(n) = 1$$

$$(\alpha * \beta)(n) = \sum_{d \mid n} \alpha(d)\,\beta\left(\frac{n}{d}\right)$$

dirichlet conv

Proof: i.e. ① $n = \sum_{d \mid n} \varphi(d)$

for every $x \in 1..n$:

$$x = x^{\text{mult}} \cdot x^{\text{coprime}}$$

$\varphi\left(\frac{n}{d}\right)$

divisor d

Then $\sum_{n=1}^{N} n = \sum_{n=1}^{N} \sum_{d \mid n} \varphi(d)$

$$\frac{N(N+1)}{2} = \sum_{k=1}^{N} \Phi\left(\left\lfloor \frac{N}{k} \right\rfloor\right)$$

$$\Phi(N) = \frac{N(N+1)}{2} - \sum_{k=2}^{N} \Phi\left(\left\lfloor \frac{N}{k} \right\rfloor\right)$$

$\Phi(N)$:

if $N \le M$ — precalc lim

return precalc[N]

if computed N before

return memo[N]

return $\frac{N(N+1)}{2} - \sum_{k=2}^{N} \Phi(\lfloor \frac{N}{k} \rfloor)$

takes only $2\sqrt{N}$ steps to compute

Complexity?

$$M + \sqrt{N} \cdot \frac{N}{M}$$

Φ computed only at divisors $\frac{N}{1}, \dots \frac{N}{k}$ until we hit $\frac{N}{k} \le M$.

$$N\sqrt{N} = M^2$$

$$N^{3/4} = M.$$

$\rightarrow$ total compl $O(N^{3/4})$

Möbius function

$$\mu(n) = \begin{cases} 0 & \text{if } n - \text{not square free} \\ (-1)^k & \text{if } n \text{ has } k \text{ prime div} \end{cases}$$

essentially used to "Inclusion-Exclusion"

(see $n^{2/3}$ algorithm)

if $f = g * \mathbb{1}$

then $g = f * \mu$.

Problem

Compute

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \gcd(i, j)$$

$$f(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} [\gcd(i,j) = k] \cdot k$$

$$= \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} k \cdot [\gcd(i,j)=k]$$

$$= \sum_{k=1}^{n} \sum_{i=1}^{\lfloor n/k \rfloor} \sum_{j=1}^{\lfloor n/k \rfloor} k\,[\gcd(i,j)=1]$$

$$= \sum_{k=1}^{n} k \cdot \sum_{i=1}^{\lfloor n/k \rfloor} \sum_{j=1}^{\lfloor n/k \rfloor} [\gcd(i,j)==1]$$

$$= \sum_{k=1}^{n} k \cdot g(\lfloor n/k \rfloor)$$

Trick

$$\sum_{d \mid n} \mu(d) = \begin{cases} 1 & n=1 \\ 0 & \text{else} \end{cases}$$

$$g(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} [\gcd(i,j)=1]$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} [\gcd(i,j)==1]$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{d \mid \gcd(i,j)} \mu(d) = \sum_{d=1}^{n} \mu(d) \left\lfloor \frac{n}{d} \right\rfloor^2$$

$2\sqrt{n}$ steps

(precalc $\sum_{i=1}^{n} \mu(i)$)

$$f(n) = \sum_{k=1}^{n} k\, g(\lfloor \tfrac{n}{k} \rfloor)$$

$2\sqrt{n}$ steps

$\Rightarrow$ total time: $O(n)$

Sum of φ

it's tempting to apply same mobius idea to $\Sigma \varphi$.

It's possible but result is less nice

$$\sum_{n=1}^{N} \varphi(n) =$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{n} [\gcd(k, n) == 1]$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{n} \sum_{d \mid \gcd(k,n)} \mu(d)$$

$$= \sum_{d=1}^{N} \mu(d) \sum_{n=1}^{N} \sum_{k=1}^{n} [d \mid k,\ d \mid n]$$

$$= \sum_{d=1}^{N} \mu(d) \sum_{n=1}^{\lfloor N/d \rfloor} \sum_{k=1}^{n} 1$$

$$= \sum_{d=1}^{N} \mu(d) \cdot \binom{\lfloor N/d \rfloor + 1}{2}$$

$2\sqrt{n}$ summands

but need to compute

$M(n) = \sum_{i=1}^{n} \mu_i$ effeciently ....

mertens function

It's possible to compute M using

$$\sum_{k=1}^{n} M(\lfloor n/k \rfloor) = 1$$