

# Ukeoppgave i uke 7, INF2440 – v2017

---

Det er foreslått følgende algoritme for å konkurrere med Eratosthenes Sil i å lage alle primtall under N raskest:

0) Alle primtallene vi finner lagres i en int [] prim = new int [(int)( N/ Mat.log(N) ) +200];

1) 2 er et primtall og legges i prim[0]

2) Vi prøver alle oddetallene c oppover (3,5,7,9,...) og:

- a. Lar c seg ikke dele med rest == 0 med alle primtall vi hittil har funnet  $\leq \sqrt{c}$  har vi funnet et nytt primtall som legges i prim[] ,  
ellers hopper vi ut av testingen av det tallet c hvis vi får en rest == 0.
- b. Når vi har nådd  $c > N$  er vi ferdig.

## Oppgave

a) Skriv et sekvensielt program som implementerer denne algoritmen.

Skriv ut alle primtall  $< 100$  for å sjekke at den er riktig. Du skal da implementere to metoder:

```
void lagPrimtallene(int n) {  
    // din kode her  
}  
  
boolean erPrimtall (int m) {  
    // denne metoden kan svare på om alle tall  
    // k < n*n er et primtall  
    // din kode her  
}
```

b) Sammenlign denne algoritmen tidsmessig med Eratosthenes Sil for  $N = 10, 100, 1000, \dots, 100\,000\,000$ . Kommenter tidsforbruket og hvorfor den ene algoritmen er så mye raskere for  $N > 1000$ .

c) Forklar hvordan du evt. vil parallellisere denne algoritmen.?

d) Prøv å utled hvilken orden  $O(N)$  denne algoritmen har.