⚠ Lærerveiledning - Primtall og effektivitet

Kurs: Python
Tema: Tekstbasert, Kryptografi
Fag: Matematikk, Programmering
Klassetrinn: 8.-10. klasse, Videregående skole

Om oppgaven

Denne oppgaven inngår i en serie om kryptografi, og er den første som omhandler moderne temaer. Oppgaven kan brukes frittstående.

Oppgaven er ikke testet på hele målgruppen, så tilbakemeldinger på nivået og egnede trinn er velkommen.

Oppgaven passer til:

Fag: Programmering, matematikk

Anbefalte trinn: 8. trinn--VG3

Tema: Kryptografi, primtall, kjøretid for algoritmer

Tidsbruk: Dobbeltime

Kompetansemål

- [ ] **Valgfag programmering**: Prinsipper som ligger til grunn for god programmeringspraksis inngår også i hovedområdet, deriblant forklaring og dokumentasjon av løsninger og programkode; vurdering og analyse av egen og andres programkode (Fra hovedområdene)
- [ ] **Valgfag programmering**: omgjøre problemer til konkrete delproblemer
- [ ] **Matematikk X**: bruke Eratostenes' såld til å finne primtall
- [ ] **Matematikk X**: bruke kongruensregning til å analysere delelighet

Forslag til læringsmål

- [ ] Elevene kan finne ut om et utvalg tall er eller ikke er primtall
- [ ] Elevene kan forstå og forklare hvorfor koden kjørere raskere eller tregere fra steg til steg

Forslag til vurderingskriterier

- [ ] Eleven oppnår middels måloppnåelse ved å fullføre oppgaven.
- [ ] Eleven oppnår høy måloppnåelse ved å kunne forklare hvilken innvirkning endringene i koden har på kjøretiden
- [ ] Eleven oppnår ekstra høy måloppnåelse ved å implementere Erostathones sil, og selv undersøke hvor store tall koden kan brukes på og fortsatt operere noenlunde raskt.

Forutsetninger og utstyr

- [ ] **Forutsetninger**: God kjennskap til Python, noe matematisk modenhet
- [ ] **Utstyr**: Datamaskin med Python installert

Fremgangsmåte

Vi har dessverre ikke noen konkrete tips, erfaringer eller utfordringer tilknyttet denne oppgaven enda.

På de laveste trinnene kan temaet kan virke matematisk krevende når en ser på det første gang. Derfor kan det kanskje være nyttig å først og fremst angripe det fra et programmeringsperspektiv, for koden i seg selv er ikke særlig komplisert. I neste omgang kan man da bruke det en har programmert for å forstå matematikken bedre.

Variasjoner

Eksterne ressurser