KUNGL TEKNISKA HÖGSKOLAN

Institutionen för Numerisk analys och datalogi

Primtal och faktorisering

Johan Håstad
Klass I
johanh@nada.kth.se

Vårt problem

Givet ett rimligt stort tal

$$\begin{aligned} N = \ & 9324894190123791048152332319394135 \\ & 4114125392348254384792348320134094 \\ & 9234732854190125435244491330224499 \\ & 9724622330012312308881872002376666 \\ & 3019134151166139518510341256153023 \\ & 2324525239230624210960123234120156 \\ & 8091041095013034986140128651 23 \end{aligned}$$

är det primtal?

Om inte, kan vi faktorisera det?

Vill ha effektiva algoritmer som kan köras inom rimlig tid på existerande dator.

Varför?

- Grundläggande fråga, redan Gauss var fascinerad av den.

- Svaret av stor praktisk betydelse inom kryptografi, RSA-systemet.

Effektiv aritmetik

Tal med n siffror.

Addition genomförs med $\approx n$ operationer.

Multiplikation med standardalgoritm tar $\approx n^2$ operationer. Kan förbättras till $\approx n \log n \log\log n$ operationer.

Primalitet?

Faktorisering?

Provdivision kräver $\geq 10^{n/2}$ operationer och är ogenförbart för $n \geq 40$. Räcker $\approx n^c$ för någon konstant c?

Polynomiell tid, **P**.

Fermats lilla sats

Sats: Om N är primtal och $1 \leq a \leq N-1$ så är resten av a^{N-1} vid division med N lika med 1.

$2^6 = 64$ ger rest 1 vid division med 7

men

$2^8 = 256$ ger rest 4 vid divison med 9.

För ett a och N så kan resten av a^{N-1} vid division med N beräknas via $\approx n$ multiplikationer av n-siffriga tal.

Gammalt faktum (1976)

Miller och Rabin beskrev en mindre utvidgning av Fermats lilla sats som avslöjar alla icke primtal genom att välja slumpvis a.

Ger en effektiv probabilistisk algoritm att testa primalitat som ger rätt svar med hög sannolikhet (t.ex. $1-2^{-100}$) för varje N.

Kostnad: $100n$ multiplikationer av n-siffriga tal.

Nytt resultat (2002)

Agrawal, Kayal, Saxena hittade en betydligt mer komplicerad utvidgning av Fermats lilla sats som *alltid* gör rätt.

Går i polynomiell tid. Första varianten i tid $\approx n^{12}$ men nu finns varianter som närmar sig n^6, (n^4 om vi tillåter slumpval men inga misstag).

Betydligt långsammare än probabilistiska tester både i praktik och teori.

Filosofisk fråga: Hjälper slump att avgöra denna typ av deterministiska frågor?

Faktorisering

Väsentliga framsteg över provdivision.

Bästa algoritm: Talkroppssållet, som använder mycket matematik, t.ex. faktorisering av algebraiska heltal.

100-siffriga tal är tämligen lätta att faktorisera, 200-siffriga utom räckhåll.

På sista 20 åren har hårdvaru-förbättringar och algoritm-förbättringar betytt lika mycket.

Slutord

Primalitet är enkelt, tusentals siffror inga problem, största kända primtal, $2^{25964951}-1$ har drygt 7,8 miljoner siffror.

Faktorisering verkar svårt men det finns inga bevis att så är fallet.

Faktorisering enkelt på kvantdatorer, kan dessa byggas?