Täljaren

— Matematiskt dividerande —

Gunnar Lindholm
gunnar @ taljaren. se

februari 2008

Täljaren är en (nästan) månatlig skrift om matematik och matematikundervisning riktad till gymnasielärare och alla andra intresserade. Jag uppmanar läsarna att höra av sig med egna funderingar och tankar eller intressanta saker som kommit upp i klassrummet eller i andra sammanhang. Det materialet publiceras i kommande nummer. Allt för att stimulera till eftertanke, kompetensutveckling och för att inte glömma bort hur kul det är med matematik.

Tidigare nummer finns alltid på
`http://www.taljaren.se/matematik.php`

1 κappa 2008

Nu är det dags igen för tävlingen för alla matematiklärare. Jag hoppas på att det blir en lika intressant tävling i år. Tävlingens hemsida är `http://www.math.su.se/kappa/2008/`. Vi får hoppas att ännu fler ställer upp i tävlingen i år.

Jag har även pratat med tävlingsjuryn och de har pratat med Theducation och de kommer göra lite reklam för det här utskicket i samband med att de skickar mail till alla som varit med i tidigare tävlingar. De skall ha ett jättestort TACK för detta!

Förhoppningsvis kan detta intressera många lärare. Du hittar många tidigare nummer på
`http://www.taljaren.se/matematik.php`

1.1 Fråga 1

Så här lyder första frågan i tävlingen.

> Fatima och Kevin bakar bullar och kakor till ett föräldramöte. De tänker sälja bullarna för 10 kr och kakorna för 5 kr som ett bidrag till klasskassan. Av Kevins pappa har de fått en och en halv liter mjöl, 100 gram smör och övriga ingredienser. Till varje bulle går det åt 7 cl mjöl och 5 gram smör och till varje kaka går det åt 4 cl mjöl och 2 gram smör. Vilket blir det maximala bidraget till klasskassan om de räknar med att kunna sälja sammanlagt högst 30 bakverk, antingen bullar eller kakor?

2 Boktips

Prime numbers av David Wells. Full med kul fakta om primtal samt satser och förmodanden om dessa.

Känner du t.ex. till Bonses olikhet? Den säger att om p_n är det n:te primtalet så gäller

$$p_{n+1}^2 < p_1 p_2 p_3 \dots p_n \qquad (1)$$

för $n > 3$. Det känns som en ganska grov olikhet i och med att när n är ganska stort, så gäller att $p_{n-1} \approx p_n \approx p_{n+1}$, och då kan högerledet i (1) grovt uppskattas

$$p_1 p_2 p_3 \dots p_{n-1} p_n > 2p_{n-1}p_n \approx 2p_{n+1}^2$$

Olikheten skall tolkas som mycket stor.

Det kan dock tilläggas att man kan göra gapen mellan primtalen godtyckligt stora, så att $p_{n+1} - p_n$ är godtyckligt stort. Låt säga att jag vill ha $p_{n+1} - p_n = 1001$, d.v.s. vi skall ha 1000 tal mellan dem och alla mellanliggande tal skall vara sammansatta. Hur hittar jag p_n och p_{n+1}? Tyvärr finns inga direkta formler, men vill man vara säker på att hitta talen mellan p_n och p_{n+1} kan vi välja dessa 1000 tal som

$$1001! + 2, 1001! + 3, \dots, 1001! + 1001$$

Alla dessa konsekutiva tal är sammansatta ty de är delbara med 2, 3, ..., 1001 respektive. Vi behöver alltså bara välja $p_n < 1001! + 2$ och $p_{n+1} > 1001! + 1001$. Det blir dock ganska stora värden då 1001! innehåller 2571 siffror.

Nu kan man fråga sig hur man beräknar $\log_{10} 1001!$. Antingen använder man Maple och kommandot

```
evalf(log10(1001!));
```

eller så beräknar man

$$\log_{10}(1001!) = \log_{10}(1 \cdot 2 \cdot 3 \dots \cdot 1001) =$$

$$\log_{10} 1 + \log_{10} 2 + \log_{10} 3 + \dots + \log_{10} 1001$$

som grovt kan uppskattas med integralen

$$\int_1^{1002} \log_{10} x \, dx = \frac{1}{\log 10} \int_1^{1002} \log x \, dx =$$

$$\frac{1}{\log 10} \left[x \log x - x \right]_1^{1002} = \frac{1002 \log 1002 - 1002 + 1}{\log 10} \approx 2572$$

Ganska skaplig approximation.

3 En bok i handen

Jag började fundera lite över vilka böcker man borde sätta i händerna på de elever som har läst åtminstone D-kursen på gymnasiet. De brukar ju bara få en bok full med räkneuppgifter. Sida upp och sida ner med uppgifter, oftast ordnade i växande svårighetsgrad. Visst är det bra att räkna uppgifter, men för att liva upp tycker jag det behövs en annan form av litteratur också. Böcker där det finns mycket text och inte bara färgglada rutor.

Att få elever att läsa sådant inser till och med jag är en hopplös uppgift. Kanske någon superpedagog som läser detta har något bra tips för hur man gör? Högläsning?

Jag tänker nu lista böckerna i den ordning jag tycker de kan läsas.

1. *Reella tal* av Ivan Niven. En mycket, mycket trevlig liten bok i Prisma-serien. Boken är lättläst (även om den kräver att man ibland har papper och penna till hands) och författaren förklarar bra.[1] [2]

2. *Fermats gåta* av Simon Singh. Det är en historisk bok, men den beskriver en stor del av vårt kulturarv och är mycket allmänbildande.

   En annan mer historisk bok är *Stora matematiker* av Harold Edwards, Ettore Picutti och Laurent Schwartz. Också ett bra tips.

3. *Elementa* av Euklides. En självklar klassiker. Jag skulle rekomendera att först lusläsa axiomen och definitionerna. Läs några av de inledande bevisen. Skumma sedan igenom kommande bevis för att få en uppfattning om vad som kommer att hända. Tag sedan tag i någon sats, följ beviset till punkt och pricka. Varje referens till en tidigare sats måste också följas upp. Läs de satser som används för att bevisa den valda satsen. Känslan av hur snyggt geometrin är uppbyggd hoppas jag infinner sig hos eleven. Eleven kan givetvis begränsa läsandet till just det jag beskriver ovan. Det bör räcka.

4. *På tal om tal* av Lars Nystedt. Helt klart en "tyngre" variant och mer omfattande än *Reella tal*, men det är en mycket läsvärd bok.

5. *e, the story of a number* av Eli Maor. En bok bara om talet *e*.

6. *Trigonometric delights* av Eli Maor. Den bjuder på trigonometri.

7. A*n imaginary tale, the story of* $\sqrt{-1}$ av Paul J Nahin. Den kommer visa dig mycket kul med komplexa tal.

8. *Introduktion till matematiken* av Tord Ganelius. En mycket trevlig bok; kräver dock en del förkunskaper.

9. *What is mathematics* av Richard Courant, Herbert Robbins och Ian Stewart. Den används som kursbok på Lunds tekniska högskola i en kurs om matematikens grunder (om jag minns rätt) och det borde placera den på en viss plats i läsordningen.

10. *Encounter with mathematics* av Lars Gårding. Den hamnar sist eftersom det är min uppfattning att man skall spara det bästa till sist. Dessutom tror jag den skulle knäcka den oförberedde.

Slutligen vill jag ta upp

- *Mathematics, from the birth of numbers* av Jan Gullberg.

[1] Jag kan rekomendera fler böcker i Prismas serie från 60-talet. T.ex. Tord Halls *Gauss* eller varför inte Ogilvy och Andersons *Talteori för alla*? Den titeln borde ju vara inbjudande.

[2] På `http://www.bokborsen.se` hittade jag i skrivande stund två exemplar av mitt standardtips som kommer först.

Gullbergs verk läses bäst genom att sitta i en bekväm fotölj, med något gott att dricka, och sedan slå upp ett avsnitt på måfå och bara läsa och njuta. Som alternativ till det kan man börja från början och gå igenom matematikens utveckling kronologiskt på ett mycket angenämt sätt. Jag längtar efter en utökad upplaga där man fyller på med allt som hänt inom matematiken fram till våra dagar. Det vore intresant att se hur man hanterar alla kopplingar mellan olika grenar, det får ju inte bara bli en samling korsreferenser.

Fler tips på böcker mottages tacksamt från läsekretsen.

4 Koden

Jag fick ett intressant problem presenterat för mig av Anders Dessmark och vi diskuterade problemet en del. Här tänker jag sammanfatta resultatet av våra ansträngningar.

Ur ett vardagsproblem går det ut på följande. Tänk dig att du skall öppna ett kodlås genom att trycka in den rätta kombinationen. Antag att koden är N tecken lång och att du har A olika symboler att välja mellan för varje tecken. I vår vardag är det alltid så att efter varje kombination man har tryckt in så måste man även trycka på en specialknapp för att signalera att man är klar med intryckningen. Då finns det ingen genväg utan du är tvungen att prova varje permutation av tecknen. Det finns totalt sett A^N olika permutationer. Vanligt vis är $A = 10$ och $N = 4$ vilket ger tiotusen kombinationer.

Nu skall vi istället tänka oss ett lite enklare kodlås där du inte behöver trycka på någon specialknapp utan du kan trycka på hur många knappar som helst och så fort den rätta koden dyker upp någonstans i strömmen av tryckningar, då öppnas låset.

Även här kan man prova sig fram med alla de A^N permutationerna, men Anders tänkte då att man måste kunna göra detta bättre.

Låt oss exemplifiera med $A = 3$ och $N = 2$. Låt symbolerna vara $0, 1, 2$ och en fullständig listning av alla tänkbara möjligheter ger oss då följande koder

```
00 01 02 10 11 12 20 21 22
```

Det är totalt sett NA^N knapptryckningar och man skulle isåfall ha tryckt in koden

```
000102101112202122
```

d.v.s. 18 tecken. Direkt bör man observera att det är ju helt onödigt att trycka tre stycken 0:or i början. Högst två 0:or i rad är av intresse för oss. Med lite funderande kommer man snart fram till att följande sekvens innehåller alla tänkbara koder.

```
0011210220
```

Övertyga dig om det på egenhand.

Längden på koden är 10 tecken och den undre gränsen för vad som är möjligt att nå är $A^N + N - 1$ som i detta fall ger $3^2 + 2 - 1 = 10$. Vi nådde precis den undre gränsen!

4.1 Undre gräns

För att förstå hur $A^N + N - 1$ kan vara en undre gräns så tänker vi oss att vi först matar in en kod, vilken som helst. Det ger oss N tecken. Nu har vi $A^N - 1$ koder kvar att mata in och det allra bästa vi kan göra är att täcka in en ny kod med varje ny symbol vi lägger på vår sekvens. Det ger då ett tillskott på $A^N - 1$ tecken och totalt $A^N + N - 1$.

Hur mycket bättre är denna undre gräns då än den naiva lösningen med alla kombinationer? Vi ser att vi har kvoten

$$\frac{NA^N}{A^N + N - 1} \approx N \tag{2}$$

mellan längderna på de två kodsträngarna. I (2) gäller det för alla praktiska fall att A^N är mycket större än $N-1$. Om din kod har fyra tecken i sig, då kommer detta sätt att gå fyra gånger snabbare i genomsnitt. Detta är ingen fantastisk uppsnabbning.

4.2 Öppna frågor

Våra diskussioner slutade med ett antal öppna frågor.

1. Är det alltid möjligt att konstruera den kortaste koden?

2. Hur ser en algoritm ut som hittar den kortaste koden?

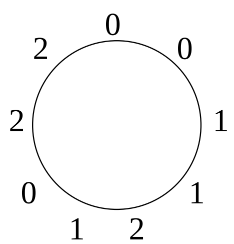
Då det gäller algoritmen så är det tyvärr inte helt lätt att komma på den. För att komma fram till en minimal kod skulle man kunna tänka sig att man börjar med en slumpvist vald kod och sedan lägger till nya koder med ett tecken åt gången. Problemet med denna algoritm är att vi kan hamna i återvändsgränder. Med tanke på det stora antalet möjliga val i varje situation så inser man att det är ohållbart att skriva ett program som provar alla tänkbara koder. Ett sådant program skulle ta allt för lång tid att köra.

Man kan formulera om problemet som ett grafproblem och det kan kanske hjälpa att finna ett svar på frågorna. Observera att vi är inte helt säkra på att de två formuleringarna är helt likvärdiga. Dock gäller att en lösning av grafproblemet ger oss en lösning till det ursprungliga problemet.

Egentligen handlar hela problemet om att få fram ett cykliskt mönster på A^N tecken. Observera att den korta koden

```
0011210220
```

blir cyklisk om vi klipper bort den inledande 0:an och sluter mönstret. Vi får mönstret som visas i figur 1.

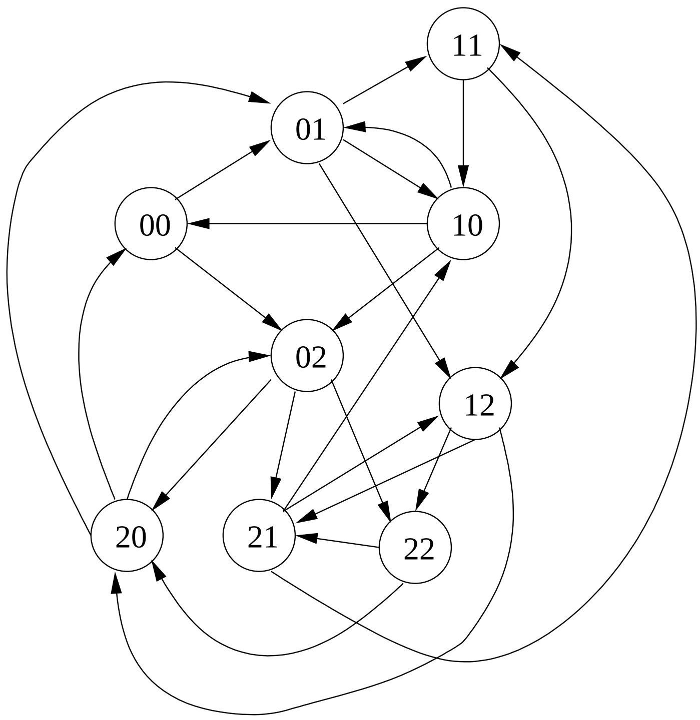


Figur 1: Koden uttryckt som ett cykliskt mönster

Man kan klippa upp cykeln var som helst och få fram en enda lång sträng om man kompleterar med de tecken som behövs för att täcka de koder som försvinner vid snittet.

För att formulera grafproblemet konstruerar en graf med en kod i varje nod, d.v.s. A^N noder. Vi placerar ut en riktad båge från noden (koden) u till noden v ifall vi får v genom att ta bort första tecknet i u och lägger till sista tecknet i v. D.v.s. från u kan vi skapa v genom att bara lägga till ett tecken. Vi tar inte med bågar som går från en nod och tillbaka till sig själv.

I vårt exempel med tre olika tecken och två tecken i koden får vi följande graf, se figur (2).



Figur 2: En förhoppningsvis fullständig graf för att beskriva kodproblemet för tre tecken 0,1,2 i en kod med bara två tecken.

4.3 Superkoden

Nu kommer nästa problem. Om vi har samma lås som vi talat om här ovan, hur gör vi i fall vi inte vet längden på koden? Då får vi börja med alla koder av längd 1, sedan alla koder av längd 2, längd 3, o.s.v..

Här observerar vi att om vi täcker in alla koder av längd N så har vi med detta även täckt in alla koder av kortare längd. Det hjälper oss dock inte alltför mycket eftersom det i praktiken skulle innebära att vi måste skapa en kod för en viss längd. Den hemliga koden skulle ju ha kunnat vara ett tecken längre. Vi skall alltså skapa en oändligt lång sträng som täcker in alla möjliga strängar så effektivt som möjligt, i någon mening. Jag är inte på det klara än med vad denna "mening" skall innebära.

Den som har svar på ovanstående frågor får gärna höra av sig.