UPPSALA
UNIVERSITET

Välkomna till

Programmeringsmetodik DV1

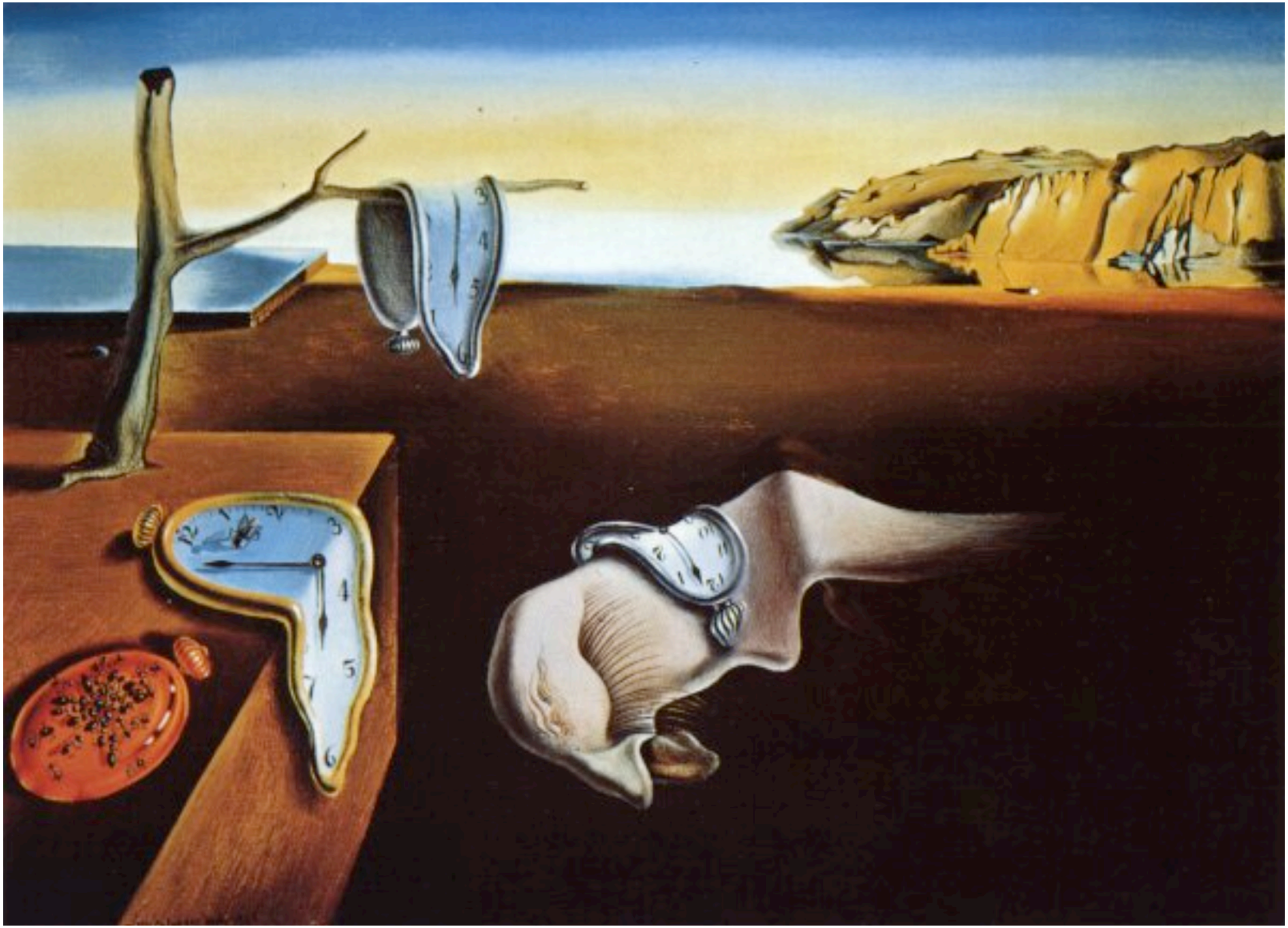
Programkonstruktion 1

http://www.it.uu.se/
edu/course/homepage/pkpm/HT06

Föreläsare

Lars-Henrik Eriksson

lhe@it.uu.se, http://www.it.uu.se/katalog/lhe

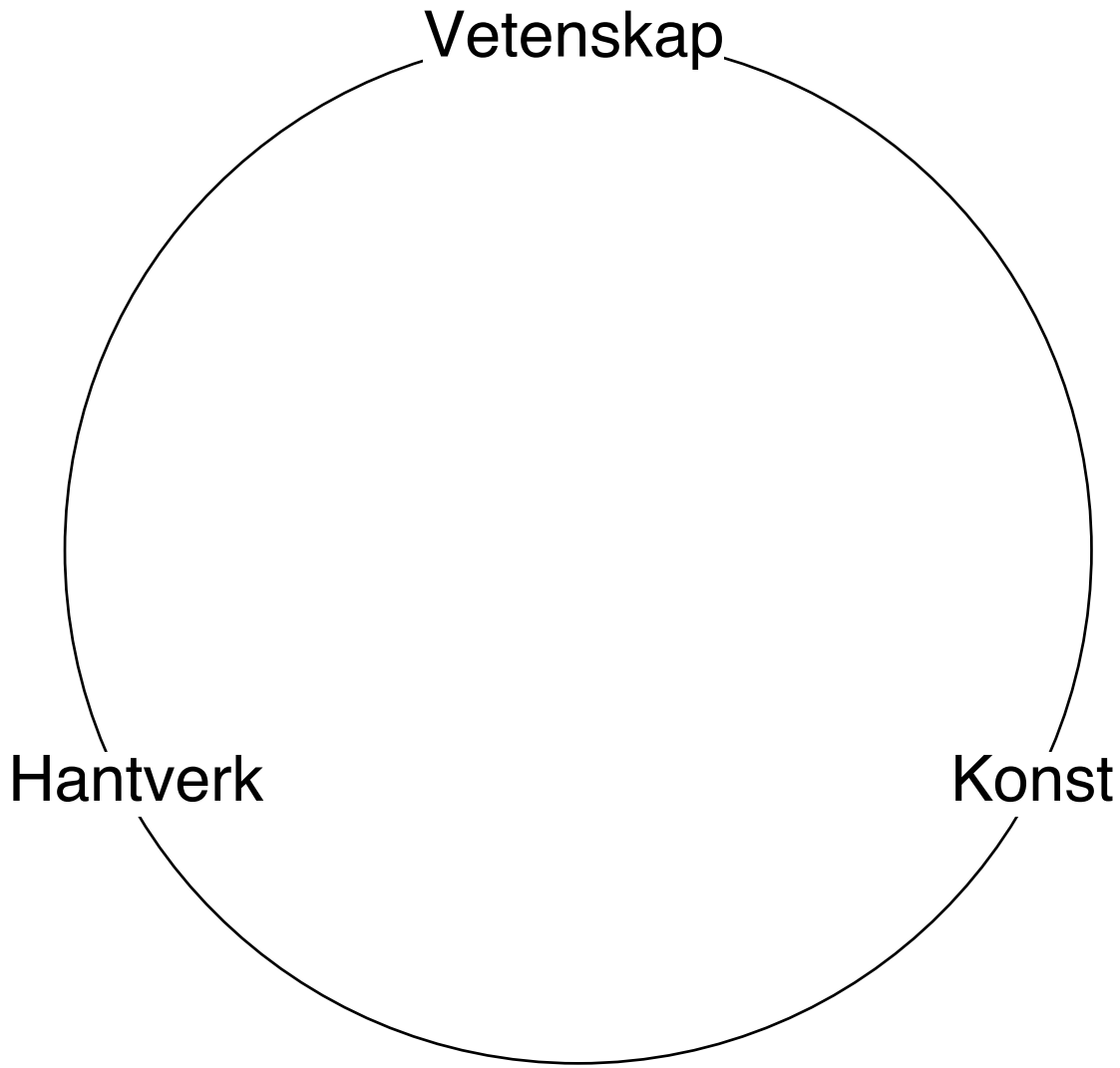


Salvador Dalí: Minnets varaktighet (1931)

Vad är ett ”program”?

- Datorn
  - kan själv inte utföra något
  - har potential för att utföra allt.
- Program är *instruktioner* till datorn hur den skall utföra en uppgift.
- Program utför *beräkningar* där det undersöker och ändrar *data*.
- Data *representerar* någonting utanför datorn själv.
- Program = *Algoritmer* + *Datastrukturer*

(Min) syn på programmering



Biff Stroganoff

Matrecept kan ses som en sorts program. De innehåller instruktioner om hur man skall utföra en uppgift.

Skär 500 g benfritt nötkött i cm-tjocka strimlor, ca 3 cm långa.
Skala och hacka 2 gula lökar.
Bryn kött och lök.
Tillsätt 1 tsk salt, 1 krm peppar, 2 msk tomatpuré och 1,5 dl vatten.
Efterstek i 5 min.
Späd med 1,5 dl crème fraiche mot slutet.

Hur detaljerade skall instruktioner vara?

Det beror på läsaren!

Hur man skär kött i strimlor

1. Tag kött, en vass kniv och en skärbräda.
2. Tag en köttbit.
3. Skär loss en smal bit kött med kniven.
4. Finns det kött kvar på köttbiten? Gå i så fall till steg 3.
5. Finns det köttbitar kvar? Gå i så fall till steg 2. Annars är det klart!

”Hur man skär...” utgör ett *underprogram* till ”Biff Stroganoff”.

Observera att instruktioner kan

— läggas i följd — ges som alternativ — upprepas —

.....de grundläggande *programstrukturerna*

Algoritm

”Effektiv procedur” för beräkning

1) Mekanisk (skall inte kräva ”intelligens” för att förstå)
2) Ändliga resurser (måste bli färdig)
3) Deterministisk (det skall alltid vara entydigt vad som skall göras)
4) ”Aritmetiserbart” problem (problemet skall kunna representeras i termer av data – egentligen heltal)

Vad som är ”mekaniskt” beror förstås på vem som tolkar algoritmen.

Ett program är (oftast) en algoritm.
(men med ”algoritmer” i dagligt tal menar man normalt fundamentala konstruktionsmönster för programmering).

Euklides från Alexandria (325 – 265 f.v.t.)

Största gemensamma delare

(gcd – Greatest Common Divisor)

Det största heltal som jämnt delar två andra heltal.

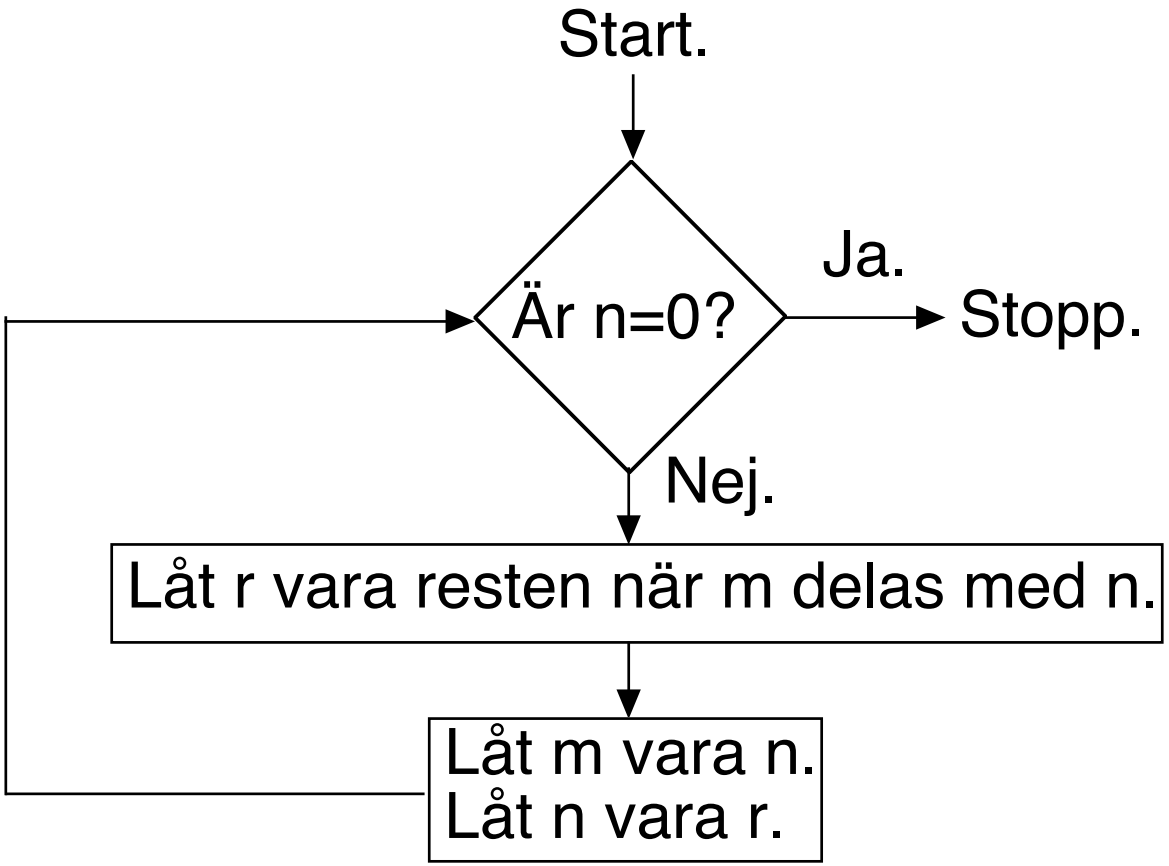
Exempel: Vad är $\frac{126}{168}$?

gcd till 126 och 168 är 42. 126 = 3*42, 168 = 4*42 så

$$\frac{126}{168} = \frac{3*42}{4*42} = \frac{3}{4}$$

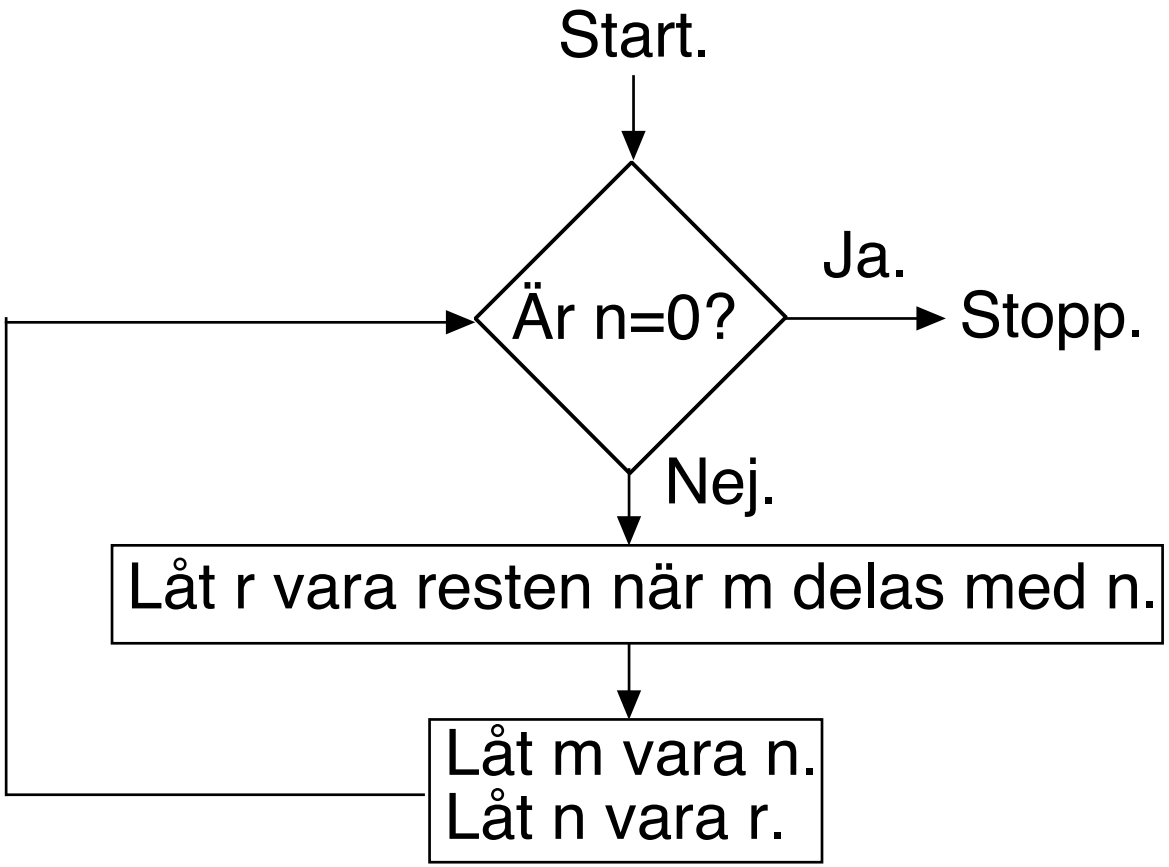
Euklides algoritm

Förvillkor: m och n är positiva heltal
Eftervillkor: m är största gemensamma delare till de ursprungliga m och n.
Variant: n



Euklides algoritm: exempel

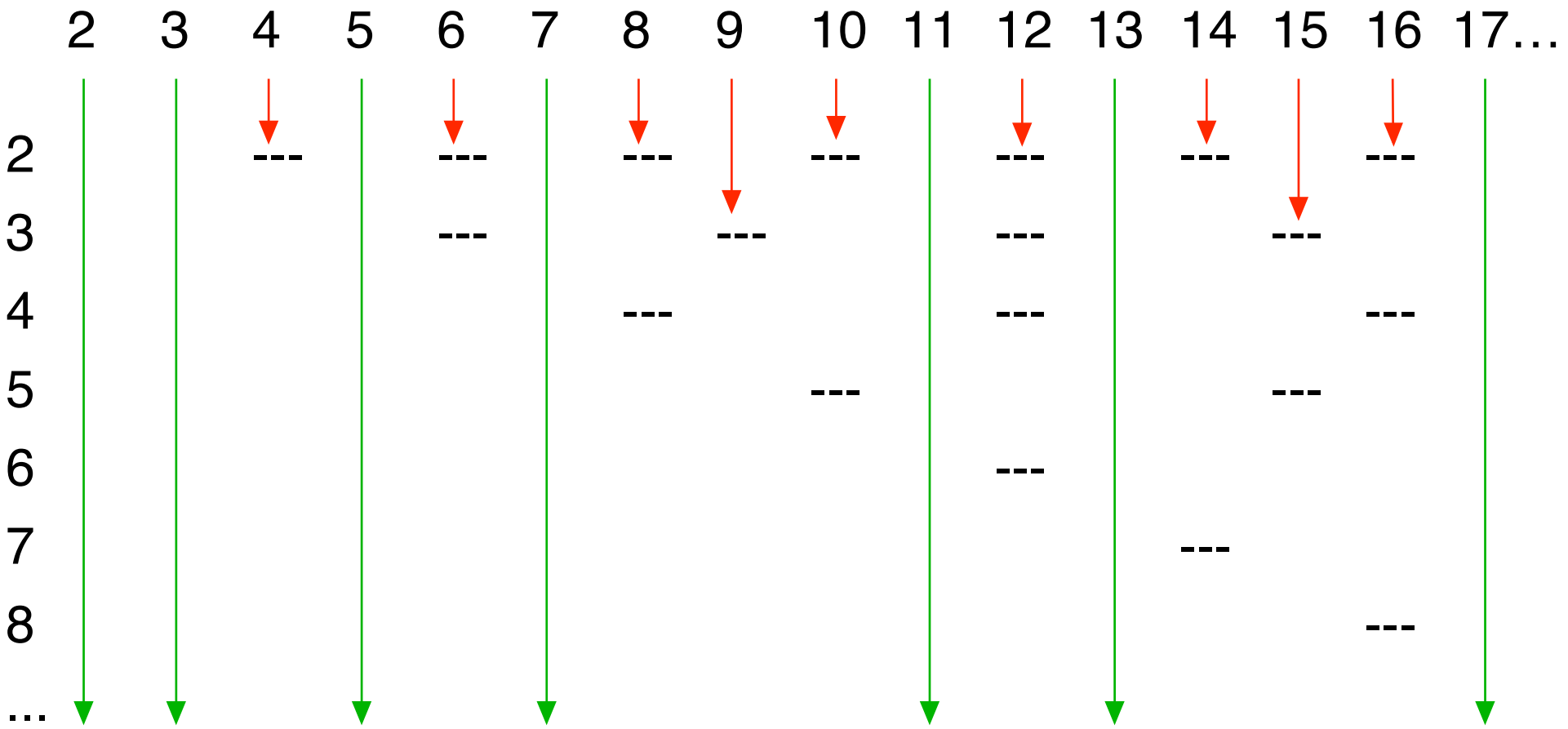
Största gemensamma delaren till 126 och 168



m	n	r
126	168	
		126
168		
	126	
		42
126		
	42	
		0
42		
	0	

Erathostenes från Kyrene (276 – 194 f.v.t.)

Erathostenes såll



De tal som faller igenom sållet är primtal.

Pseudoalgoritmer

Eratostenes såll är en *pseudoalgoritm*. (En mekanisk procedur som inte uppfyller kraven på att vara en algoritm.)

Alla program är inte algoritmer.
Ofta vill man inte att program skall avslutas – t.ex. program som styr processer av olika slag:

- trafikljus
- telefonväxlar
- DVD-spelare
- kärnkraftverk...

Abu Ja’far Mohammad Ibn Musa al-Khwarizmi

(c:a 780 – c:a 850)

Två böcker av al-Khwarizmi

Efter romarrikets fall skedde en vetenskaplig nedgång i Europa. Den islamska kulturen bevarade och utvecklade grekisk vetenskap. Matematikern al-Khwarizmi verkade i Bagdad som under 800-talet var ett kulturellt och vetenskapligt centrum.

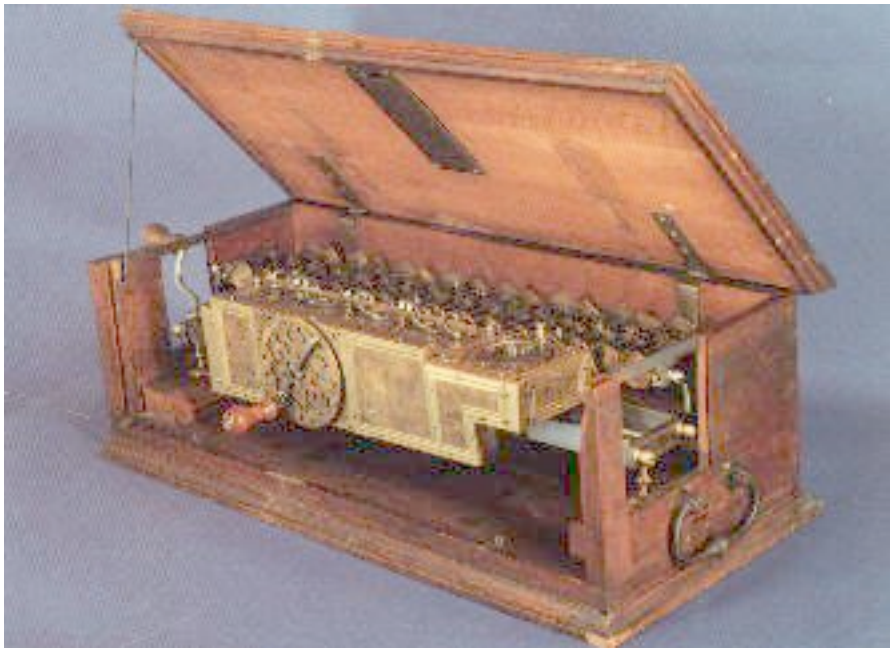
- *Om beräkning genom ”al-jabr” och ”al-muqabala”*
  Metoder för att förenkla ekvationer och lösa andragradsekvationer.
- *al-Khwarizmi om de indiska talen* (”Algoritmi de numero Indorum” – Endast en latisk översättning finns bevarad.)
  Om siffrorna 0-9 och positionssystem med siffran 0.

Förvrängning av ”al-jabr” och al-Khwarizmis namn har gett oss orden ”algebra” och ”algoritm”.

Gottfried Wilhelm von Leibniz
(1646 – 1716)

”……calculemus” (låt oss räkna)

Leibniz kalkylator



Leibniz kalylator förverkligar (*implementerar*)
algoritmer för de fyra räknesätten.

Gammal matematisk tabell

sin 0°—45°

Grader	0	1	2	3	4	5	6	7	8	9	—
0,	0,0000 17	0017 18	0035 17	0052 18	0070 17	0087 18	0105 17	0122 18	0140 17	0157 18	0175
1,	0175 17	0192 17	0209 18	0227 17	0244 18	0262 17	0279 18	0297 17	0314 18	0332 17	0349
2,	0349 17	0366 18	0384 17	0401 18	0419 17	0436 18	0454 17	0471 17	0488 18	0506 17	0523
3,	0523 18	0541 17	0558 18	0576 17	0593 17	0610 18	0628 17	0645 18	0663 17	0680 18	0698
4,	0698 17	0715 17	0732 18	0750 17	0767 18	0785 17	0802 17	0819 18	0837 17	0854 18	0872
5,	0,0872 17	0889 17	0906 18	0924 17	0941 17	0958 18	0976 17	0993 18	1011 17	1028 17	1045
6,	1045 18	1063 17	1080 17	1097 18	1115 17	1132 17	1149 18	1167 17	1184 17	1201 18	1219
7,	1219 17	1236 17	1253 18	1271 17	1288 17	1305 18	1323 17	1340 17	1357 17	1374 18	1392
8,	1392 17	1409 17	1426 18	1444 17	1461 17	1478 17	1495 18	1513 17	1530 17	1547 17	1564
9,	1564 18	1582 17	1599 17	1616 17	1633 17	1650 18	1668 17	1685 17	1702 17	1719 17	1736
10,	0,1736 18	1754 17	1771 17	1788 17	1805 17	1822 18	1840 17	1857 17	1874 17	1891 17	1908
11,	1908 17	1925 17	1942 17	1959 18	1977 17	1994 17	2011 17	2028 17	2045 17	2062 17	2079
12,	2079 17	2096 17	2113 17	2130 17	2147 17	2164 17	2181 17	2198 17	2215 18	2233 17	2250
13,	2250 17	2267 17	2284 16	2300 17	2317 17	2334 17	2351 17	2368 17	2385 17	2402 17	2419
14,	2419 17	2436 17	2453 17	2470 17	2487 17	2504 17	2521 17	2538 16	2554 17	2571 17	2588
15,	0,2588 17	2605 17	2622 17	2639 17	2656 16	2672 17	2689 17	2706 17	2723 17	2740 16	2756
16,	2756 17	2773 17	2790 17	2807 16	2823 17	2840 17	2857 17	2874 16	2890 17	2907 17	2924
17,	2924 16	2940 17	2957 17	2974 16	2990 17	3007 17	3024 16	3040 17	3057 17	3074 16	3090
18,	3090 17	3107 16	3123 17	3140 16	3156 17	3173 17	3190 16	3206 17	3223 16	3239 17	3256
19,	3256 16	3272 17	3289 16	3305 17	3322 16	3338 17	3355 16	3371 16	3387 17	3404 16	3420
20,	0,3420 17	3437 16	3453 16	3469 17	3486 16	3502 16	3518 17	3535 16	3551 16	3567 17	3584
21,	3584 16	3600 16	3616 17	3633 16	3649 16	3665 16	3681 16	3697 17	3714 16	3730 16	3746
22,	3746 16	3762 16	3778 17	3795 16	3811 16	3827 16	3843 16	3859 16	3875 16	3891 16	3907
23,	3907 16	3923 16	3939 16	3955 16	3971 16	3987 16	4003 16	4019 16	4035 16	4051 16	4067
24,	4067 16	4083 16	4099 16	4115 16	4131 16	4147 16	4163 16	4179 16	4195 15	4210 16	4226

Tabellen ger värden på sinus för vinklar i steg av 0,1°

Charles Babbage
(1791 – 1871)

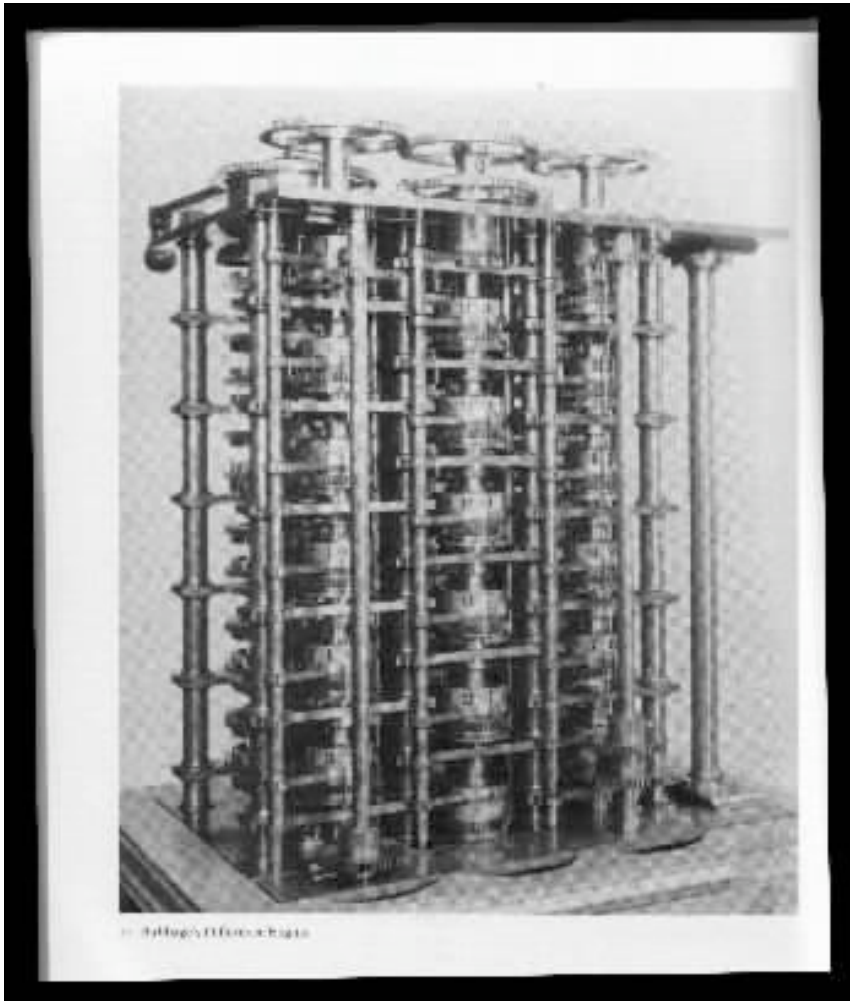
Differenskalkylen

Polynom kan approximera matematiska funktioner som sinus, logaritm etc. (med taylorutvecklingar – se kurs i analys)

Differenskalkylen beräknar polynom med enbart additioner.

x	$2x^2-3x+4$	1:a diff.	2:a diff	3:e diff.
0	4			
		-1		
1	3		4	
		3		0
2	6		4	
		7		0
3	13		4	
		11		?? (0)
4	24		?? (4+0)	
		?? (11+4+0)		
5	?? (24+11+4+0)			

Babbages differensmaskin (del)



Differensmaskinen implementerar en algoritm för differenskalkyl.

Augusta Ada Byron grevinna av Lovelace

(1815 – 1852)

Babbages analytiska maskin

Den analytiska maskinen skulle *programmeras* att utföra *godtycklig* algoritm.

Beståndsdelarna i princip samma som i en modern dator, men allt byggt mekaniskt.

Ada Lovelace skrev det första *programmet* i modern mening – för beräkning av Bernoullital.

Den analytiska maskinen kunde aldrig byggas – dåtidens finmekanik räckte inte.

Alonzo Church (1903 – 1995)

Alan Mathison Turing (1912 – 1954)

Vad är ”mekaniskt”?

Matematiska modeller av beräkningar.

Church: λ-kalkylen (”lambdakalkylen”)
Turing: turingmaskiner

Church-Turings tes: Dessa modeller kan beskriva *allt* som går att beräkna.

I praktiken: uttryckbart som program i en dator.

Alla beräkningar kan inte mekaniseras enligt Church/Turing.
Kan de då utföras alls?

λ-kalkylen har blivit ett viktigt verktyg inom datavetenskapen.

Turing award

Datavetenskapens ”nobelpris” är uppkallat efter A.M. Turing och delas sedan 1966 ut av Association for Computing Machinery(ACM)

Några pristagare:

- 1983: Ken Thompson, Dennis M. Ritchie – operativsystemet UNIX.
- 1988: Ivan Sutherland – grundläggande principer för datorgrafik
- 1991: Robin Milner – programspråket ML (bl.a.)
- 2001: Ole-Johan Dahl och Kristen Nygaard – objektorienterad programmering (Simula)
- 2002: Ronald L. Rivest, Adi Shamir, Leonard M. Adleman – kryptografi med publika nycklar (”RSA-krypto”).
- 2003: Alan Kay – objektorienterad programmering (Smalltalk) och ideerna bakom persondatorer.
- 2004: Vinton G. Cerf och Robert E. Kahn – Internet (TCP/IP)

Datarepresentation

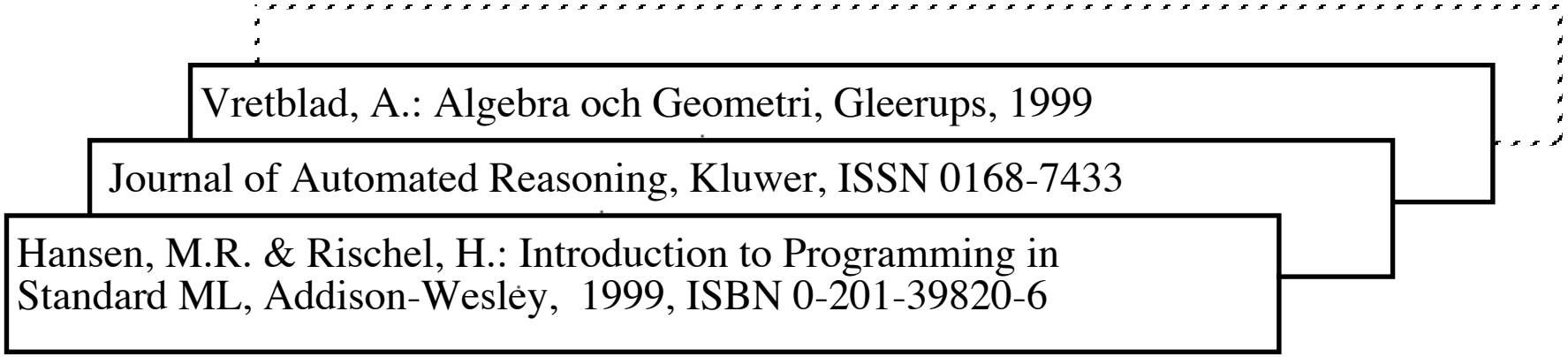
Datorer är byggda för att arbeta med *heltal*. Med hjälp av dessa kan annan information *representeras*. (Minns ni kravet på algoritmer om *aritmetiserbarhet*?)

T.ex. kan text representeras genom att *koda* bokstäver:
genom kodningen A=1, B=2 etc.kan ordet ”UPPSALA”
representeras som talföljden 20 15 15 18 1 11 1.

En bild kan byggas upp av *bildpunkter* (pixels) av bestämd färg.
Färger kan kodas som tre tal som anger inblandning av rött, grönt och blått i procent. T.ex. kan mörkgult kodas som 100 75 0.
En bild represeneras då som en (stor) mängd sådana taltripplar.

Datastrukturer

En bibliotekskatalog kan representeras som en *upprepning* av *poster*



Varje post *sätts samman* av olika *fält* – här texter (*strängar*) och tal.
Poster kan ha *olika utformning* beroende på vilka data som behövs.

Sammansättning, upprepning och alternativ är de grundläggande metoderna för strukturering av data. (Jämför programstrukturer!)

Ada Lovelace om datarepresentation

"Again, [the *Analytical Engine*] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine . . . Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."

Notes on the description of the Analytical Engine, 1843

Grace Murray Hopper
(1906 – 1992)

”If it's a good idea . . . go ahead and do it.
It is much easier to apologize than it is to get permission”

Högnivåspråk

Datorn styrs med numeriska koder (*maskinspråk*) vars funktion definieras i termer av datorns inre uppbyggnad.

Data representeras ytterst som numeriska koder.

Människor vill ha mer lättanvänd notation. Därför finns *användarorienterade programspråk* (*högnivåspråk*) för att konstruera program.

En *kompilator* är ett program översätter från ett högnivåspråk till maskinspråk.

Grace Hopper skrev den första kompilatorn (≈ 1952).

Programmeringsmetodik/konstruktion är mer än att bara skriva program...

Ni skall även:

- lära er ett högnivåprogramspråk (Standard ML)
- förstå betydelsen av
  - specifikationer (vad?)
  - motivering (varför?)
  - dokumentation i övrigt
  - elegans, precision, klarhet...
- få en första introduktion till
  - algoritmer och datastrukturer
  - algoritmkomplexitet

Små- och storskalig programmering

På denna kurs skall ni lära er *småskalig* programmering – att själv skriva små program, oftast för eget bruk. (”Små” betyder här några tusen rader programkod.)

En helt annan sak är *storskalig* programmering – att många programmerare tillsammans utvecklar stora program för någon uppdragsgivare. Stora program kan bestå av miljontals rader programkod. För detta krävs inte bara erfarenhet utan även andra kunskaper som behandlas på kurserna Programmeringsmetodik 2 och Programvaruteknik.

Jämför med att laga mat hemma till sig själv eller familjen mot att vara kock i ett restaurantkök eller vid en festmiddag för hundratals gäster!

Thomas Thorild
(1759 - 1808)

”Tänka fritt är stort men tänka rätt är större”

Devis över ingången till universitetsaulan

En kurs på universitetet…

En universitetsutbildning skiljer sig från andra utbildningar genom att

- vila på *vetenskaplig grund*. Det som lärs ut skall inte vara fritt tyckande utan vila på ”vetenskaplig … grund samt på beprövad erfarenhet.” (högskolelagen)
- skall ge förståelse för inte bara *hur* utan även *varför*.
- skall leda till *självständigt och kritiskt tänkande*. (högskolelagen)
  Vad har ni för anledning att tro eller inte tro på det som jag säger?

Man får spekulera i att homeopati fungerar eller att program-utvecklingsmetod X ger billiga program av hög kvalitet, men i verkligheten är det underbyggda påståenden som gäller.
Det kostar liv och egendom att ”tänka fritt” vid brobygge, behandling av sjukdomar, konstruktion av programvara…

Kursupplägg

Föreläsningar:	A-föreläsningar (grundläggande). B-föreläsningar (fullständiga – inkl. A-materialet) C-föreläsningar (överkurs)
Lektioner:	Redovisning/genomgång av laborationer, lärarledda övningar (första 28/9)
Laborationer:	Självständiga övningar på kursmaterialet ensam eller i *grupper om två*, med schemalagda handledningstider (första 29/9)
Inlämnings-uppgifter:	Större uppgifter som löses självständigt och individuellt.Inlämning i december resp. januari.
Tentamen:	Torsdagen den 21/12.

Hur man blir godkänd på laborationerna

1) Lämna in en (nästan) riktig lösning i tid – normalt kl. 08:00 andra arbetsdagen efter den schemalagda labben (första gången den 3/10) – *eller*

2) Lämna in ett *seriöst försök* till lösning *i tid* – i detta fall får du betyg ”K” (komplettering) på labben – *och*
   - Deltag på följande lektion (i detta fall alltså *obligatoriskt*!)
   - Lämna in en (nästan) riktig lösning senast vid nästa inlämningstid.

I annat fall blir laborationen *underkänd*.

För att få godkänt på kursen måste *7 av de 10* laborationerna vara godkända.

Presentation och opposition

För att träna på *muntlig framställning* och *kritiskt tänkande* skall alla vid något tillfälle under kursen

- Presentera sin lösning på en labbuppgift på lektionen.
- Opponera på någon annans presentation.

Opposition – opponenten läser lösningen i förväg och funderar: Är detta rätt? Varför är det gjort som det är gjort? Kan det göras på bättre sätt? Efter presentationen ställer opponenten frågor om detta och den som presenterar (respondenten) skall kunna svara på frågorna.

Assistenterna väljer före varje lektionstillfälle ut vilka som skall presentera och opponera – ni får alltså reda på det med rätt kort varsel.

Vilken var webbadressen nu igen?

http://www.it.uu.se/edu/course/homepage/pkpm/HT06

Läs informationen om kursen!

Kontrollera vilken lab/lektionsgrupp du tillhör!

Läs nyheter på webben! Minst två gånger i veckan.