

Ej blot til lyst: Programmering og matematisk dannelse i det 21. århundrede

Henrik Kragh Sørensen

Institut for Naturfagernes Didaktik
Københavns Universitet

Konference om Programmering og Koder
Danmarks Matematiklærerforening og Forlaget MATEMATIK
Odense Congress Center, 29. september 2017

<http://www.matematikhistorie.dk/>

Nogle (tilfældige) indlæg på www.folkeskolen.dk etc.

- “I Estland lærer de at programmere fra 1. klasse”
- “Ny portal kommer lærere med it-skræk til hjælp”
- “Lærerne anbefaler: Her er programmerne, som løfter vores undervisning”
- “Vi skal have flere piger til at programmere!”

Oplæggets struktur

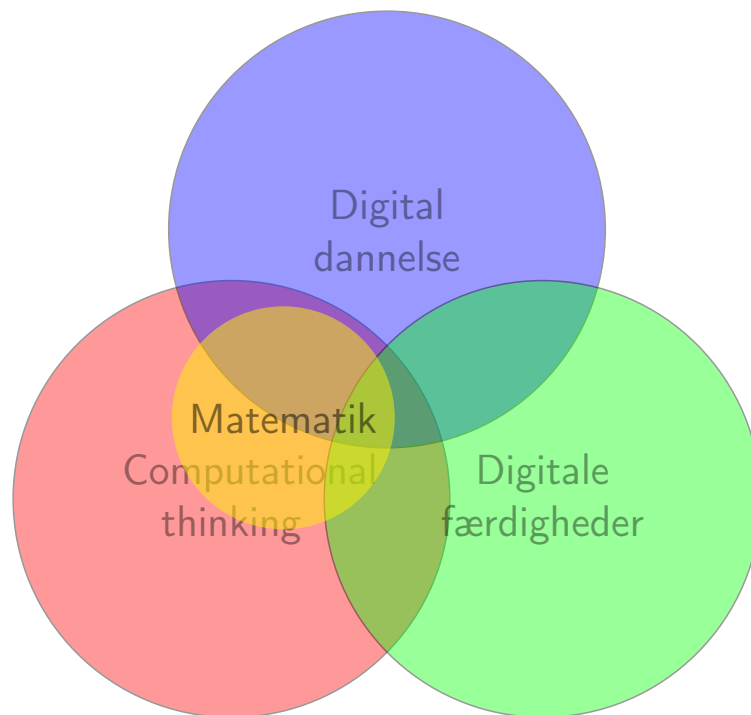
- Del 1: Matematisk dannelse i undervisningen
- Del 2: Fem forskellige 'matematiksyn'
- Del 3: Hvad er det, vi skal programmere?
- Del 4: 'Computational thinking'
- Del 5: Algoritmer og matematik
- Del 6: Kodning og programmering
- Del 7: Konklusioner og perspektiver



M. W. Johansen og H. K. Sørensen (2014). *Invitation til matematikkens videnskabsteori*. København: Forlaget Samfundslitteratur.

Del I

Matematisk dannelse i undervisningen



Del II

Fem forskellige 'matematiksyn'

Matematiksyn

Formål med matematikundervisning

Autoritær

Formålet er at tillære basale matematiske færdigheder, "numeracy" (talundskab) og social opdragelse i lydighed og autoritetstro.

Industricentreret

Formålet er at lære basale færdigheder og lære at løse praktiske problemer med matematik og informationsteknologi.

Matematikcentreret

Formålet er at opnå forståelse og evner i avanceret matematik samt en vis værdsættelse af matematikken.

Elevcentreret

Formålet er at opnå selvtillid, kreativitet og selvudtrykkelse gennem matematik.

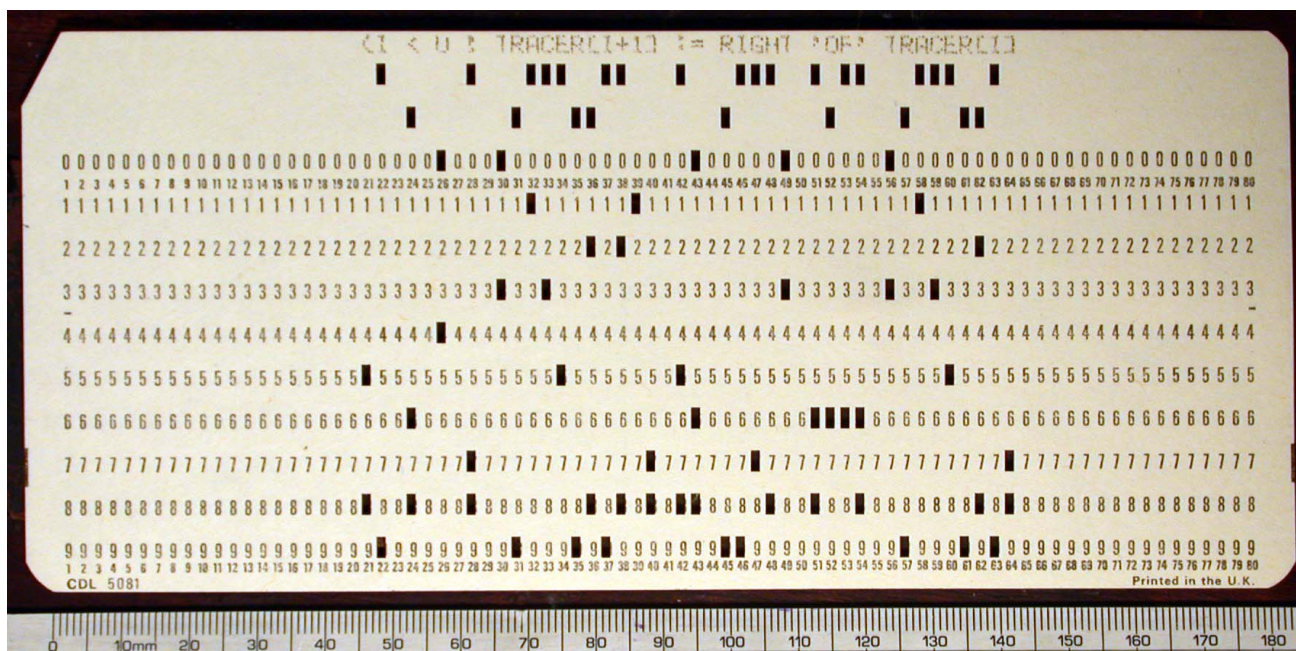
Ligestillingscentreret

Formålet er at bemyndige den lærerende som kritisk og matematisk bogligt dannet borger i et moderne samfund.

Del III

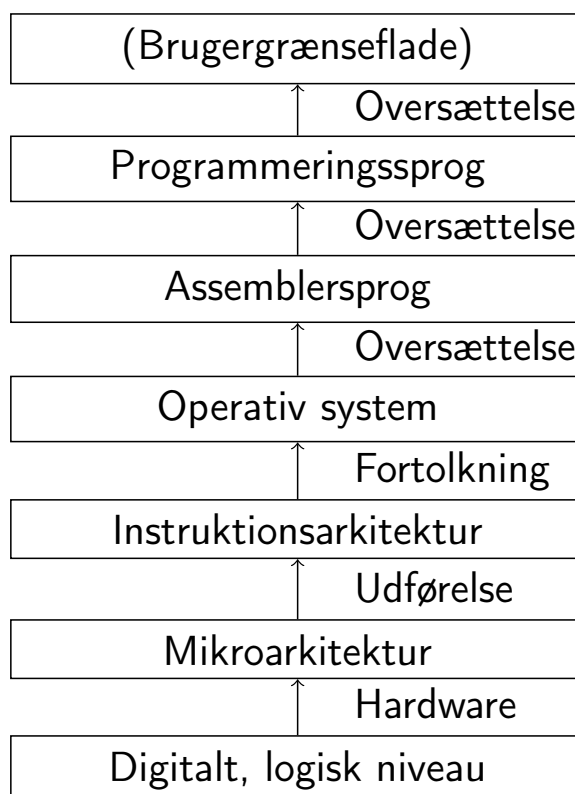
Hvad er det, vi skal programmere?

Computerens 'forældre': Hulkort og Turing



Hulkort, Hollerith og IBM

Lag på lag af abstraktioner



Del IV

'Computational thinking'

'Computational thinking'

- 'Computational thinking' som fundamental egenskab for alle
- Forbundet til andre former for modellering og problemløsning
- Systematiseret problemløsning baseret på beregninger
- Hvad er menneskers og computers relative styrker i problemløsning? Hvordan tager man hensyn til det?
- Hvordan kan man gøre 'svære' problemer 'lette'? Hvilke trade-offs følger med?
- Rekursiv tænkning, parallel tænkning, flow-control, reduktion, simulering, søgning, fejlfinding, ...
- Data og information; tal og tekst

J. M. Wing (mar. 2006). "Computational Thinking". *Communications of the ACM*, bd. 49, nr. 3, s. 33–35.

- "Turing's Men [sic!]"

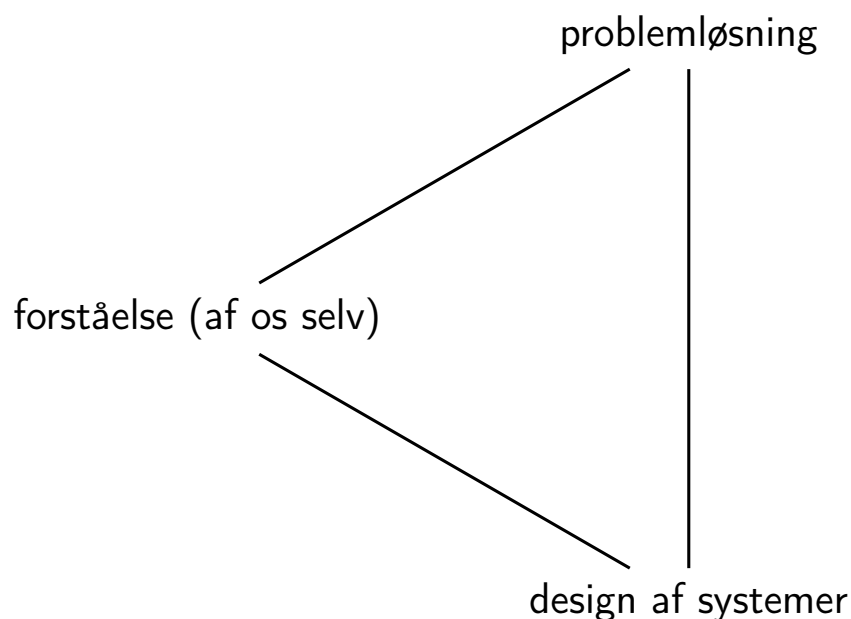
J. D. Bolter (1984/1986). *Turing's Man: Western Culture in the Computer Age*. London etc.: Penguin Books.

Men 'Computational thinking' er ...

- Conceptualizing, not programming
- Fundamental, not rote skill
- A way that humans, not computers, think
- Complements and combines mathematical and engineering thinking
- Ideas, not artifacts
- For everyone, everywhere

J. M. Wing (mar. 2006). "Computational Thinking". *Communications of the ACM*, bd. 49, nr. 3, s. 33–35.

'Computational thinking'



Del V

Algoritmer og matematik

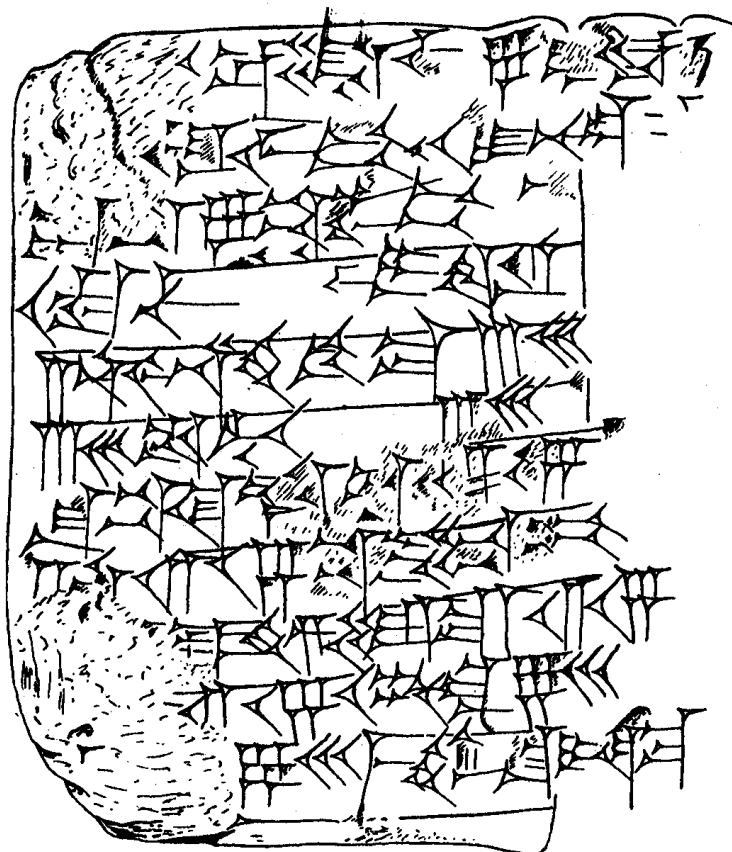
Babylonske positionstalsystemer

| | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|--|
| | 10 | | 20 | | 30 | | 40 | | 50 | | |
| 1 | | 11 | | 21 | | 31 | | 41 | | 51 | |
| 2 | | 12 | | 22 | | 32 | | 42 | | 52 | |
| 3 | | 13 | | 23 | | 33 | | 43 | | 53 | |
| 4 | | 14 | | 24 | | 34 | | 44 | | 54 | |
| 5 | | 15 | | 25 | | 35 | | 45 | | 55 | |
| 6 | | 16 | | 26 | | 36 | | 46 | | 56 | |
| 7 | | 17 | | 27 | | 37 | | 47 | | 57 | |
| 8 | | 18 | | 28 | | 38 | | 48 | | 58 | |
| 9 | | 19 | | 29 | | 39 | | 49 | | 59 | |

YBC 6967 (Yale Babylonian Collection)



YBC 6967 (Yale Babylonian Collection)



YBC 6967 som et eksempel

Igibûm overstiger igûm med 7. Igûm og igibûm hvad? Du: 7 hvormed igibûm overstiger igûm, til to bræk: 3;30. 3;30 sammen med 3;30 lad holde: 12;15. Til 12;15 som fremkommer for dig, 1 00 fladen tilføj: 1 12; 15. Ligesiden af 1 12;15 hvad? 8;30. 8;30 og 8;30, dets modstykke, indtegn. 3;30, det som holder, fra den ene udgiv, til den anden tilføj. Den ene bliver 12, den anden bliver 5. 12 er igibûm, 5 er igûm.

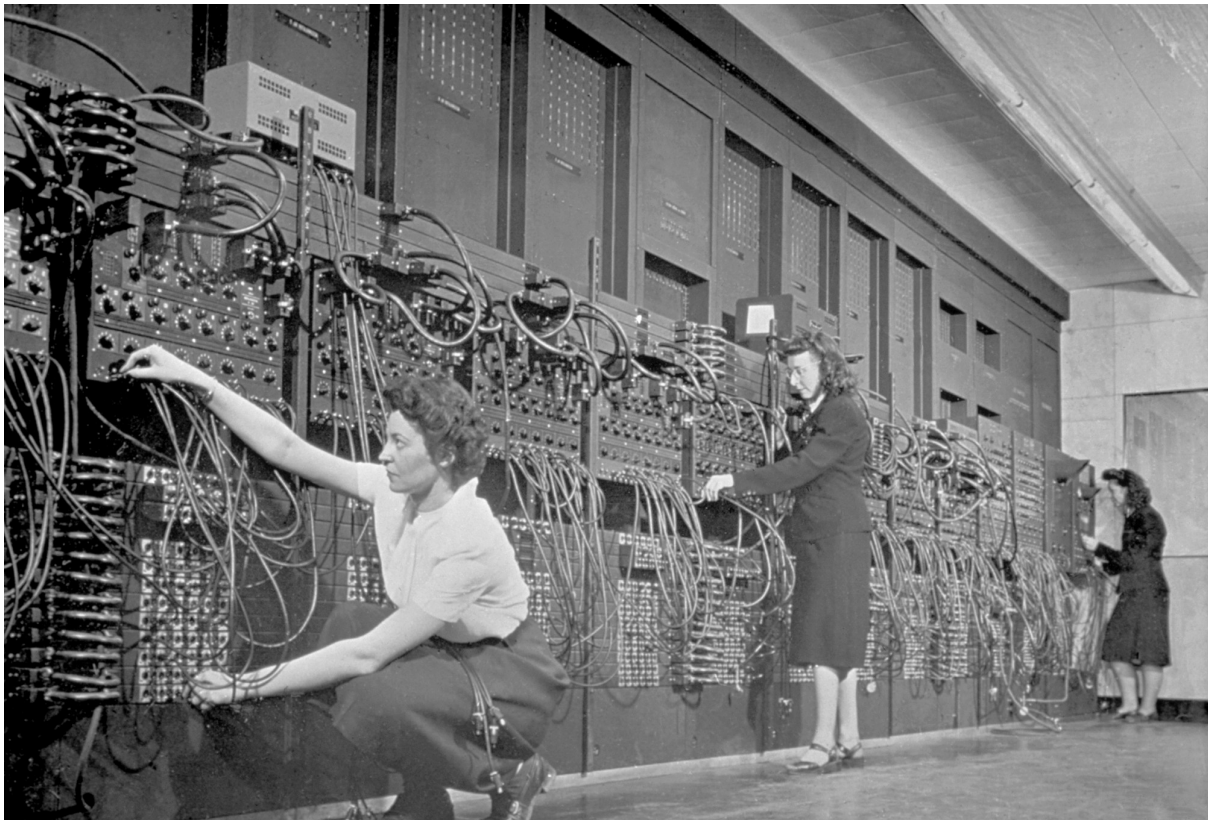
$$\sqrt{\frac{7}{2} \times \frac{7}{2} + 60} \pm 3\frac{1}{2}$$
$$\begin{cases} x - y = 7 = d \\ xy = 60 = p. \end{cases} \implies x = \sqrt{\left(\frac{d}{2}\right)^2 + p} \pm \frac{d}{2}.$$

J. Høyrup (1998). *Algebra på lertavler*. Matematiklærerforeningen.

Del VI

Kodning og programmering

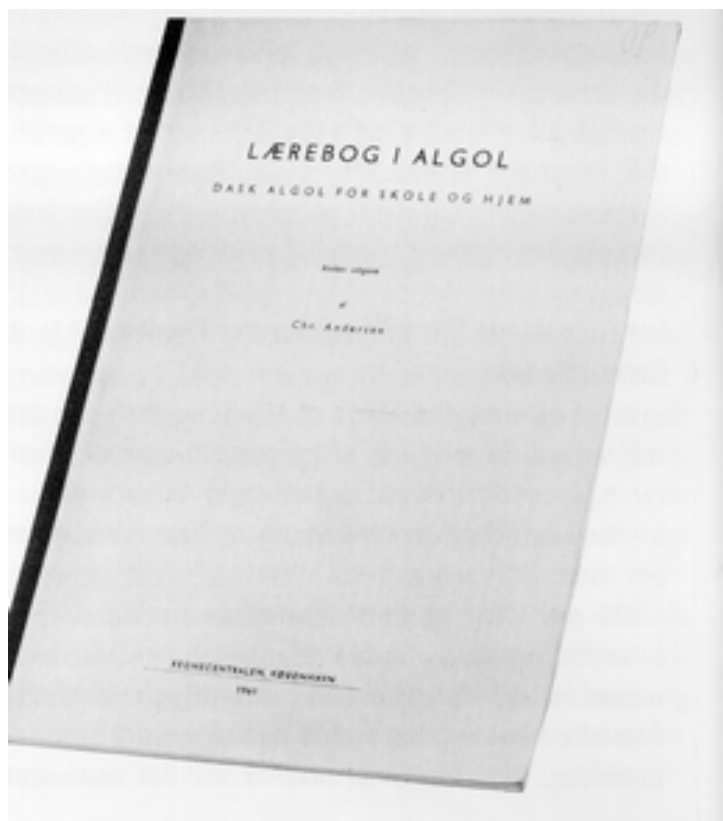
Kodere: kvinder og nørder



GIER for kodere og kongelige



Kan vi lære noget af vores egen historie?



Del VII

Konklusioner og perspektiver

- Hvis vi stiler efter en måde at træne matematisk tænkning på, så skal vi måske fokusere på at implementere algoritmer, fx til ligningsløsning eller simple grafteoretiske eksempler.
- Hvis vi stiler efter (tværfaglig) modellering og problemløsning, så kan robotter være rigtig sjove. Men vi risikerer, at vi ikke får kigget ind i den sorte kasse.
- Hvis vi stiler efter digital dannelse, så kunne adgang til store datamængder (fx Danmarks Statistik eller forskellige API'er) være relevant — men det kan måske bedre dækkes i (samspil) med andre fag?
- Hvis vi stiler efter digitale færdigheder, så er alle stykker software jo eksempler — men eksempler skal også vælges med omhu. Og igen skal vi passe på ikke at træne i at bruge lukkede, sorte kasser.
- 'Gamification' er en super motivationsfaktor — men vi skal tænke os om og gøre det inkluderende.

Nogle referencer



Bolter, J. D. (1984/1986). *Turing's Man: Western Culture in the Computer Age*. London etc.: Penguin Books.



Høyrup, J. (1998). *Algebra på lertavler*. Matematiklærerforeningen.



Johansen, M. W. og H. K. Sørensen (2014). *Invitation til matematikkens videnskabsteori*. København: Forlaget Samfundslitteratur.



Wing, J. M. (mar. 2006). "Computational Thinking". *Communications of the ACM*, bd. 49, nr. 3, s. 33–35.