



Computational thinking

Hvad er det? Hvad er det ikke? Og hvordan kan det bruges?

Anders B. Larsen, pædagogisk konsulent hos  *VidenCenter*

Hvad er computational thinking (CT)?

- Cuny-Snyder-Wing definition af **computational thinking (CT)**:

“Computational Thinking is the thought processes involved in formulating **problems** and their **solutions** so that the solutions are represented in a **form** that can be effectively carried out by an **information-processing agent**.”
[Wing2010]

- Definitionen er ret standard, men ikke operationel og stadig under udvikling.



Hvad er computational thinking (CT)?

Computational thinking (CT) omfatter:

- **Dekomposition**
- **Abstraktion** (mønsteridentifikation, modellering)
- **Algoritme** (design, automatisering)
- **Evaluering** (debugging, iteration)
- **Generalisation**



Svarer lidt til
Analyse og fortolkning
i f.eks. dansk

Algoritmedesign er central i CT
og er med til at adskille CT fra f.eks. *mathematical thinking, critical thinking* og *engineering*



CT er en problemløsningsmetode

- CT er en **problemløsningsmetode**, hvor løsningen er en **algoritme** *i bred forstand* (program, bevis, software system, trinvis proces ...), som i princippet skal kunne automatiseres.

Cuny-Snyder-Wing definition af CT:

“Computational Thinking is the thought processes involved in formulating **problems** and their **solutions** so that the solutions are represented in a **form** that can be effectively carried out by an information-processing agent.”

[Wing2010]

- CT er derfor *ikke* en generel problemløsningsmetode.

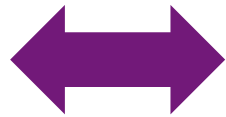
Hvad er en algoritme?

- En **algoritme** er egentligt bare en **kogebogsopskrift** (i snæver forstand):

Input

Ingredienser

Algoritme



Opskrift

Output

Mad



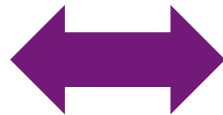
Hvad er en algoritme?

- En algoritmes **delelementer** er:

Sekvens

Først blandes vand, gær og salt, **derefter** tilsættes mel, **derefter**...

Forgrening



Hvis brødet koldhæves, **så** tilsættes 8 g gær. **Ellers** tilsættes 20 g gær.

Løkke

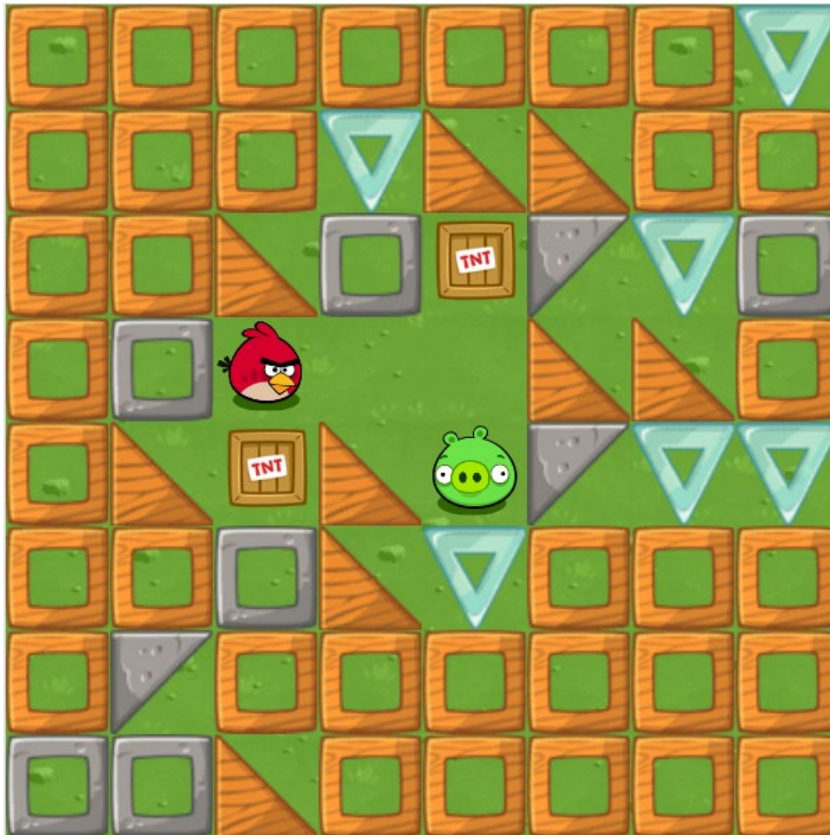
Brødet bages, **indtil** det er sprødt og gyldent (while-løkke).

- Øv dig i algoritmer med Classic Maze.



Hvad er en algoritme?

En sekvens:



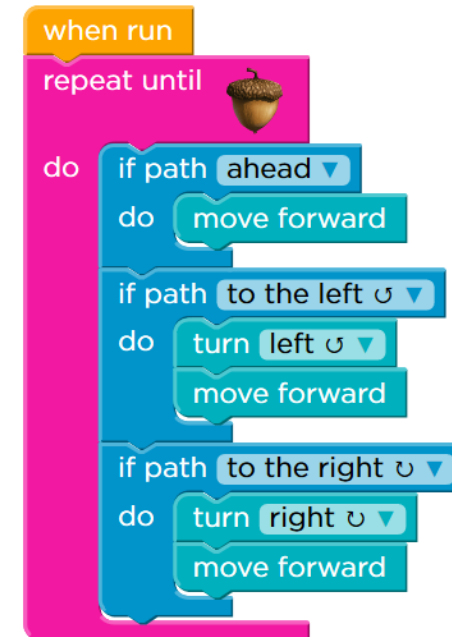
Hvad er en algoritme?

En while-**løkke** (udenom en sekvens):



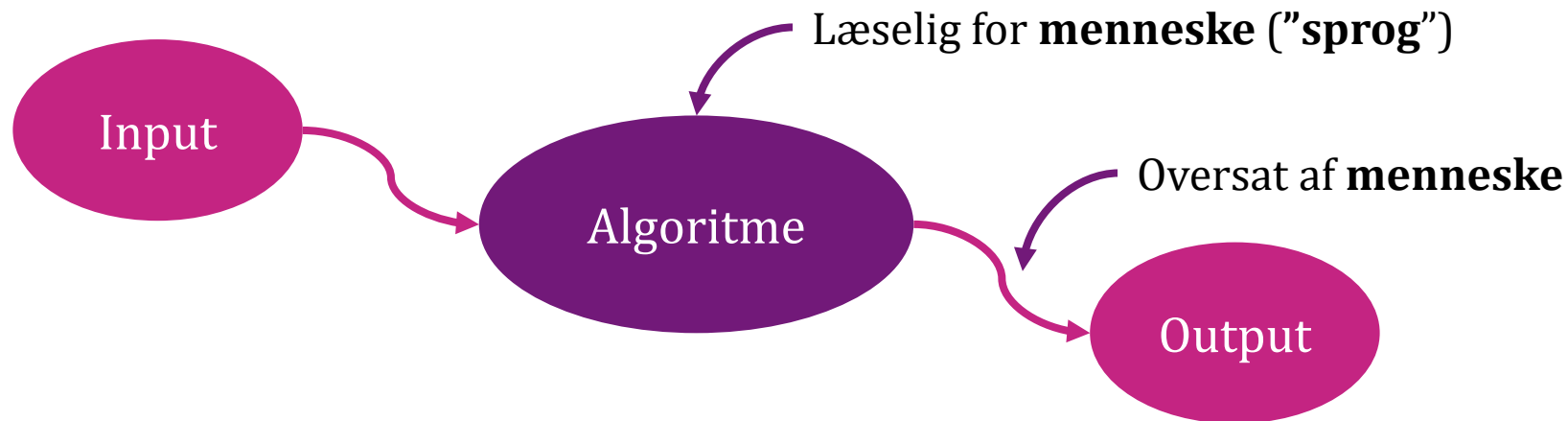
Hvad er en algoritme?

En **forgrening** (inde i en while-løkke):



CT er *ikke* programmering

- CT forudsætter *ikke* programmering eller brug af en computer.



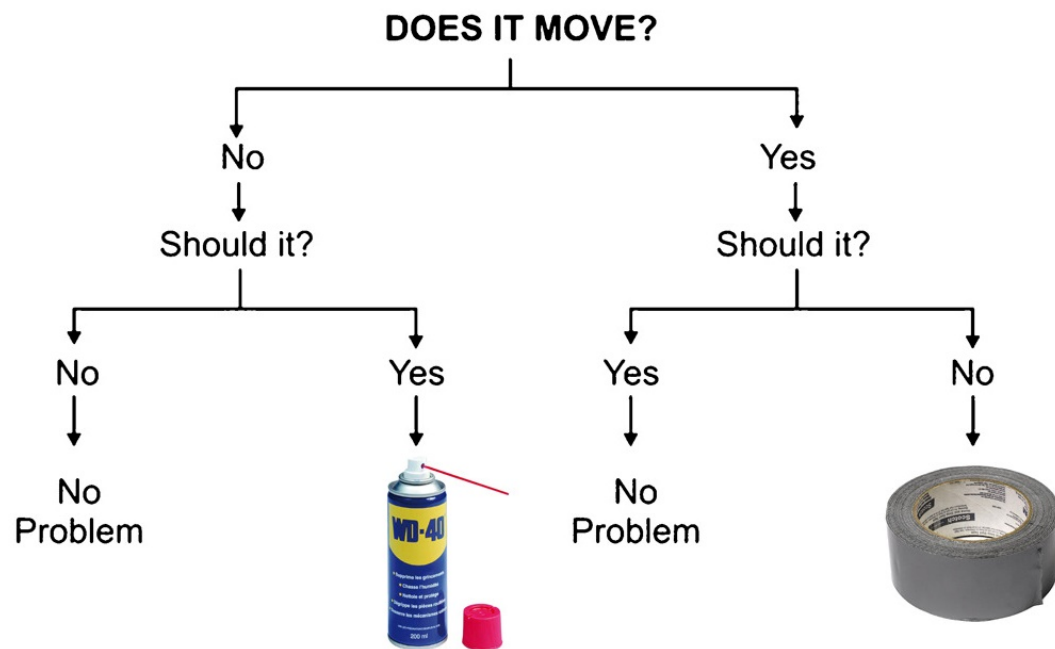
Cuny-Snyder-Wing definition af CT:

"Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an **information-processing agent**."

[Wing2010]

Algoritme uden kode... men hvordan?

Flowchart (rutediagram)

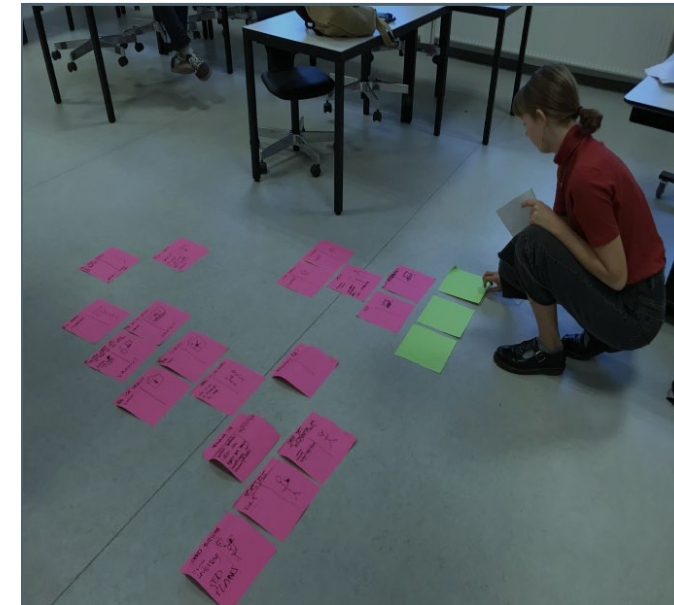
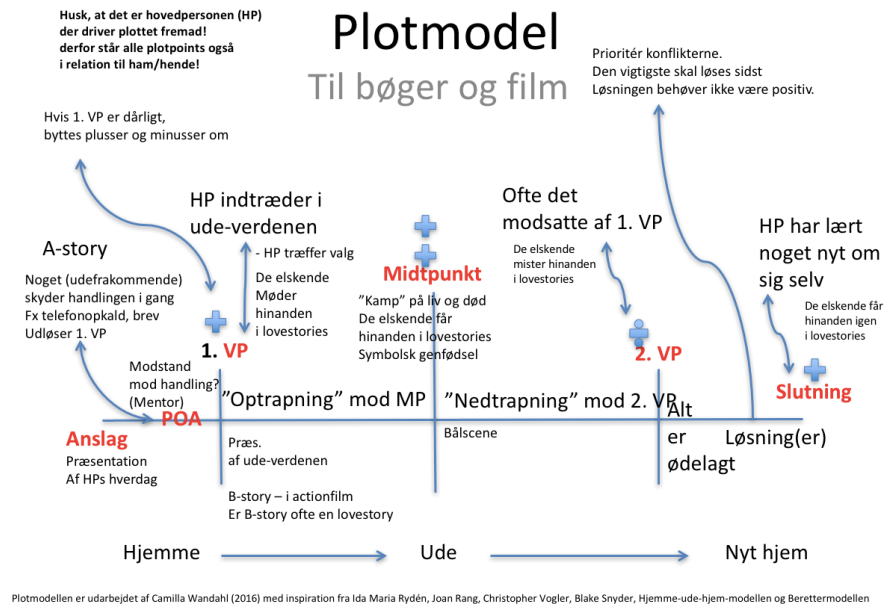


Pseudokode

- Tag postkassenøglen.
- Åbn hoveddøren.
- Gå ud til postkassen.
- Lås postkassen op.
- **Hvis** der er post i postkassen, **så** tages dette op, og postkassen låses. **Ellers** låses postkassen.
- Gå ind ad hoveddøren.
- Luk hoveddøren.
- Læg nøglen på plads.

Inspiration: Interaktive dilemmaspil i mediefag

Michael Møller udarbejdet et CT forløb om dilemmaspil i mediefag:



Dekomposition og abstraktion

Identifikation og nedbrydning af egne fortællingers betydningsbærende dramaturgiske elementer vha. plotmodel.

Algoritme

Flowcharts for "dramaturgiske veje" (forgreninger) og implementering med grafik og video i WordPress (automatisering).

Inspiration: Algoritmer i kemi og dansk

- Anne Boie Johannesson har udarbejdet et tværfagligt CT forløb i kemi og dansk:
 - Start: Simple øvelser i **algoritmisk tankegang** med **pseudokode**, f.eks. tømning af postkasse.
 - Kemi: Design af algoritmer for **mængdeberegningsopgaver** i grupper og afprøvning af andre gruppers algoritmer.
 - Dansk: Design af algoritme for udarbejdelse af **personkarakteristikker** i forbindelse med novelleanalyse.
- Algoritmisk systematisering af arbejdsgange bidrog positivt til elevernes læring.



Hvis jeg nu gerne vil kode...



- Visuelt "blok"-programmeringssprog
- Intuitivt, lettilgængeligt og brugt verden over
- Ny HTML5 version med LEGO Mindstorms EV3/WeDo 2.0 og micro:bit integration er på trapperne



- Robot med motorer, sensorer og Bluetooth.
- Visuel "blok"-programmering



- Robot med motorer, sensorer, USB, WiFi og Bluetooth
- Visuel "blok"-programmering
- Konkret og håndgribelig virkeliggørelse af CT
- Allerede brugt af en del IT undervisere



- "Lille computer" med sensorer, LED lys, knapper, USB og Bluetooth brugt verden over
- Visuelt "blok"-programmering (Blocks) eller tekstbaseret programmering (JavaScript/Python)

Hvis jeg nu gerne vil kode...

GeoGebra

- Matematik software med mulighed for tekstbaseret programmering (GGBScript og JavaScript)
- Allerede brugt af mange matematikundervisere/elever

Og alle de andre...

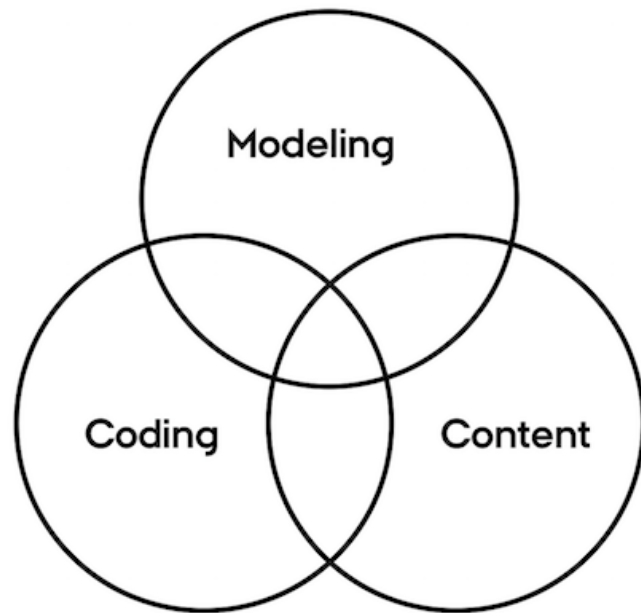


Hvis jeg nu gerne vil kode...



Inspiration: CT i gymnasiefag

- Center for Computational Thinking and Design (CCTD) har samarbejdet med 9 gymnasier i Region Midtjylland om at udvikle, afprøve og evaluere 6 korte CT-undervisningsforløb (samfunds-fag, biologi/bioteknologi og kemi).
- Undervisningsforløbene er bygget over CMC-modellen:



- **Content:** Tilegne sig fagligt stof.
- **Modeling:** Bruge, forstå, analysere samt bedømme og evaluere computerbaseret model (skrevet i NetLogo) med relation til fagstof, f.eks. reaktionshastighed i kemi.
- **Coding:** Forstå og modificere computerkode bag model, samt skrive ny kode (*use-modify-create*).



Etik, dannelse og CT?

- CT leder *ikke* nødvendigvis til etiske/kritiske (selv)refleksioner eller (digital) dannelse.
- Etik og dannelse i CT kræver rammesætning.
 - Michael Møllers dilemmaspilforløb tog udgangspunkt i FN's verdensmål om bæredygtig udvikling og gruppearbejde.
 - Anne Bois har tænkt at afslutte sit CT forløb med algoritmer i dansk og kemi med en diskussion af algoritmernes betydning og konsekvenser (med Facebook og Google som eksempler).
- Så... Er der en humanist til stede?

Advarsel: Facemash

Mark Zuckerbergs **Facemash** (rekonstruktion!):



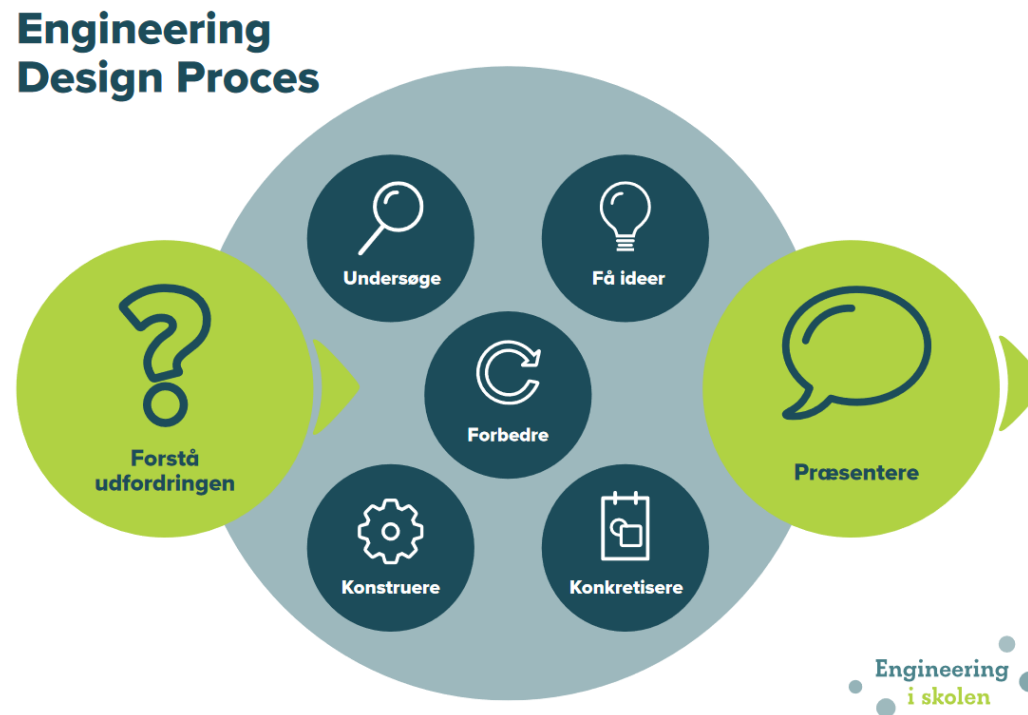
"I'm a programmer and I'm interested in the algorithms and math behind it"
- Zuckerberg

Kreativitet, innovation og CT?

- Et CT-problem kan være et **lukket problem** (med én løsning), som kan løses vha. **standardalgoritmer**.
 - Eksempel: Find det største tal i en given talrække.
- CT er derfor *ikke* nødvendigvis en kreativ eller innovativ proces.
 - Kreativitet: At skabe en ny idé.
 - Innovation: At føre en ny idé ud i livet.
- Kreativitet/innovation i CT kræver rammesætning.
 - Micheal Møller startede CT forløbet om dilemmaspil med idégenerering, som ledte til et nyt multimedieprodukt.

Inspiration: Engineering

- **Idégenerering** (kreativitet) og **prototypeudvikling** (innovation) er en del af *engineering* tilgangen.
- I folkeskoleregi er der blevet en didaktik:



Papert, Resnick og CT

Seymour Paperts "inversionsprincipper" (nogle af dem):

- **"What comes first, using it or 'getting it'?"** (Power principle)
- **"Project before Problem"**
- **"Object before Operation"** (Thingness principle)



Mitchel Resnicks fire P'er:

- **Project**
- **Peers**
- **Passion**
- **Play**

Rammeværk til facilitering af CT

- **CT concepts**
- **CT practices**
- **CT perspectives**



CT i fag og på tværs af fag

**Hvordan kunne computational thinking bruges
i dit fag og i samarbejde med andre fag?**

Tak for opmærksomheden!

Kilder

- [Shute2017] [*Demystifying computational thinking*](#), Valerie J. Shute, Chen Sun og Jodi Asbell-Clarke, Educational Research Review, 2017.
- [Wing2010] [*Computational Thinking: What and Why?*](#), Jeannette M. Wing, Link Magazine, 2010.
- [Caspersen2013] [*Computational Thinking and Practice — A Generic Approach to Computing in Danish High Schools*](#), Michael E. Caspersen og Palle Nowack, Proceedings of the Fifteenth Australasian Computing Education Conference, 2013.
- [Lockwood2017] [*Computational Thinking in Education: Where does it fit? A systematic literary review*](#), James Lockwood og Aidan Mooney, fra arxiv.org, 2017.
- [Lockwood2018] [*Computational Thinking in Secondary Education: Where does it fit? A systematic literary review*](#), James Lockwood og Aidan Mooney, International Journal of Computer Science Education in Schools, Jan 2018, Vol. 2, No. 1.
- [Lu2009] [*Thinking about computational thinking*](#), James J. Lu og George H.L. Fletcher, SIGCSE Bull. 41, 1, March 2009, 260-264.
- [Jones2011] [*The Trouble with Computational Thinking*](#), Elizabeth Jones, online artikel, 2011.

Kilder (fortsat)

- [Resnick2014] [*Give P's a chance: Projects, peers, passion, play*](#), Mitchel Resnick, online artikel, 2014.
- [Papert1996] [*An Exploration in the Space of Mathematics Educations*](#), Seymour Papert, online artikel, 1996.
- [Auener2008] [*Engineering i skolen - hvad, hvordan, hvorfor*](#), Suzie Auener, Peer Schrøder Daugbjerg, Keld Nielsen og Martin Krabbe Sillasen, udgivet online af VIA University College i samarbejde med Engineerthe Future, Astra og Naturvidenskabernes Hus, 2008.
- [Johannesson2018] [*Algoritmer og faglige metoder*](#), Anne Boie Johannesson, online artikel fra gymnasieskolen.dk, 23. jan 2018.
- [CCTD2018] CT i Gymnasiefag, online rapport offentliggjort på cctd.au.dk, april 2018.
- [Brennan2012] [*New frameworks for studying and assessing the development of computational thinking*](#), Karen Brennan og Mitchel Resnick, online artikel, 2012.