

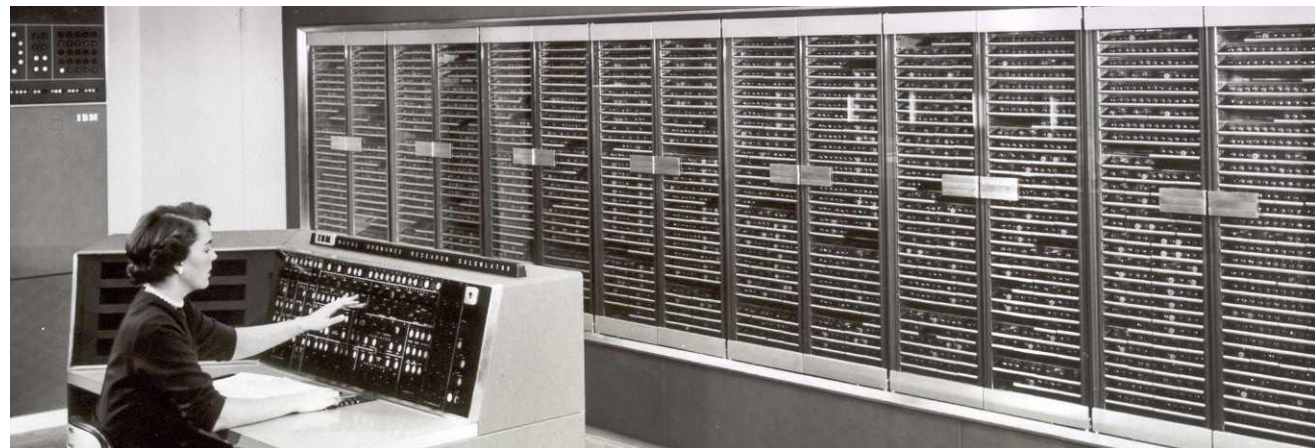
Utvikling av computational thinking (algoritmisk tenkning) som kompetanse hos teknologer og ingeniører – hvorfor er det viktig, og hvordan kan vi gjøre det?



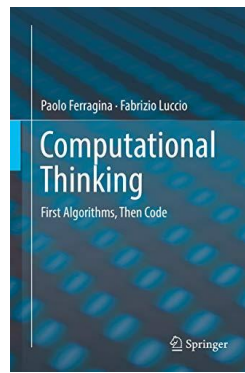
Computational Science & ingeniørkunst

- To engineer:**
1. Design and build
 2. Skilfully arrange for something to occur

Oxford ENGLISH Dictionary



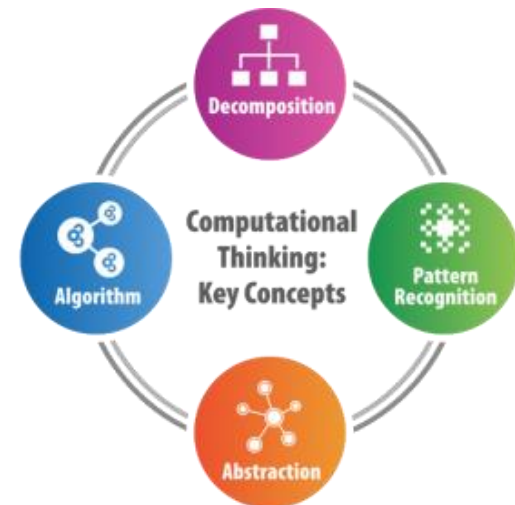
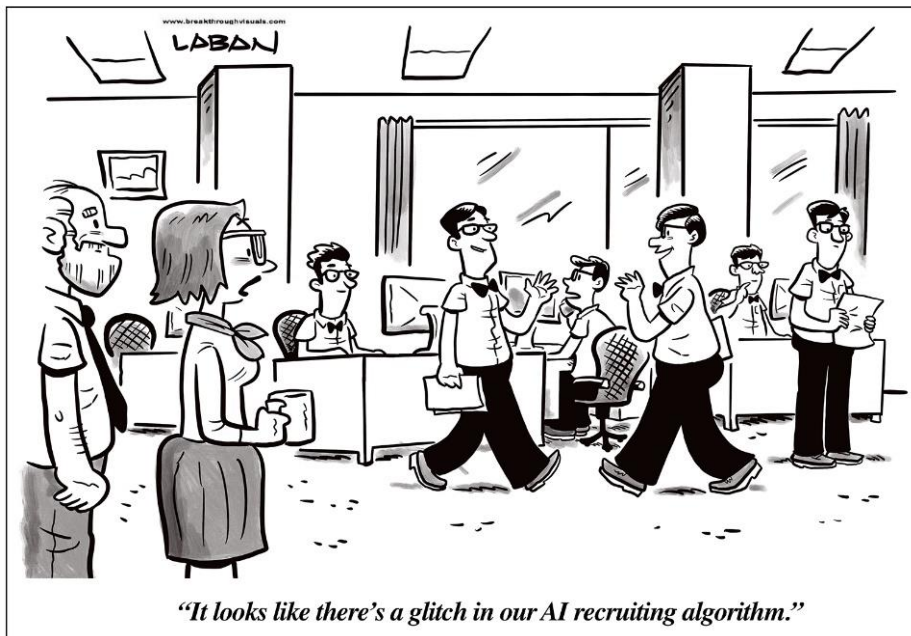
Algoritmisk tænkning – Computational thinking



Problem solving method (*actually an algorithm/recipe*):

- decompose the problem into smaller manageable sub problems
- identifying the right abstractions so as to deal with scale and complexity
- finding existing patterns or models that can be adapted (*math or pigs*)
- building an algorithm to solve the problem (*code / protocol*)
- analysing the solutions to identify the one that best meets the given situation

From cspathshala.org/



From cspathshala.org/

Algoritmisk tænkning – Computational Sciences

Today all engineering work involves computational thinking



Is it just implicit?

Computational thinking as a practice is central to all sciences and not just computer science. There is hardly any profession where computers and computational methods are not used.



All engineering educations involve computational methods

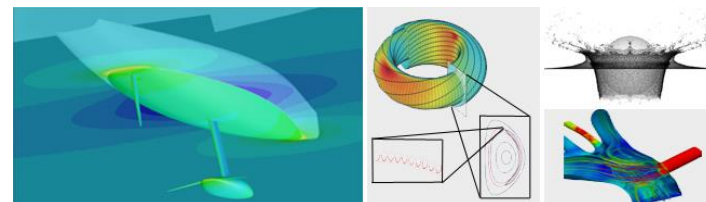
All engineering courses should involve computational thinking

All fields have tools

AND design thinking

Uddannelse Ia – choices to be made

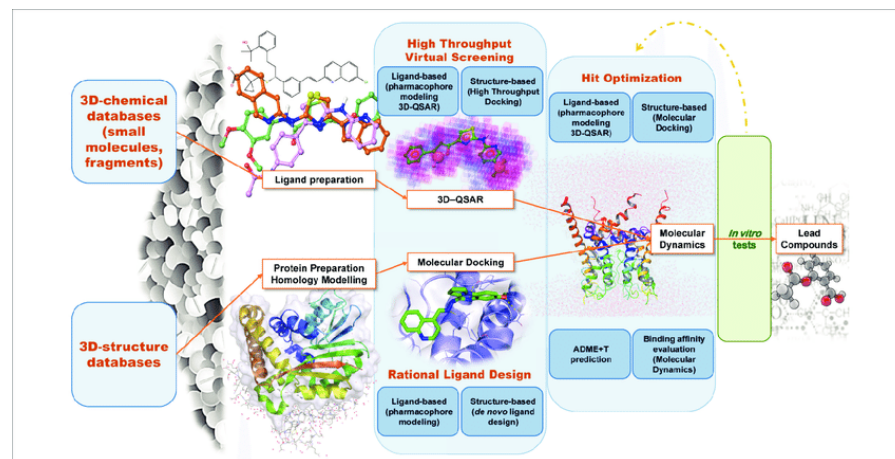
Computational Science as a tool/skill



algorithm, iteration, convergence, approximation

Use in a design context

Curriculum development needed



```
3 import os
4 import simpleknn
5 from bigfile import BigFile
6
7 if __name__ == "__main__":
8     trainCollection = 'toydata'
9     nimages = 2
10    feature = 'f1'
11    dim = 3
12
13    testCollection = trainCollection
14    testset = testCollection
```

Programming / Coding as a tool/skill

Uddannelse Ib – choices to be made

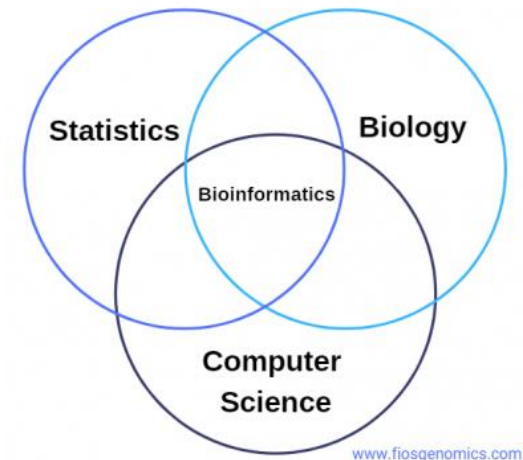
Fadel: Statistics

Data generation (simulation)



Data analytics

and merging



Hav respekt for fagenes egenart –
og vælg de faglige emner og de ingeniørmæssige eksempler med omhu



Possibilities, courses



Generiske kurser; one size fits all, med cases frit fra diverse områder

Python for beginners

Understøtter den brede, polytekniske tilgang – studenterne fremmedgøres????

Generiske basiskurser med cases/projekter fra programspecifikke fagområder

Simulation of dynamical systems and Matlab programming

Giver bredt samarbejde i organisationen – svært at implementere

Metodeorienterede kurser specielt for hvert fagligt program

Finite Element kursus for maskiningeniører

Skill eller tool – skal man kode?

Værktøj og metoder integreret i kurser rettet mod anden faglighed

Noget om fluid-flow simuleringer i et kursus om indeklima

Tool og evt skill – learning outcomes?

Uddannelse II – ever persistent challenges

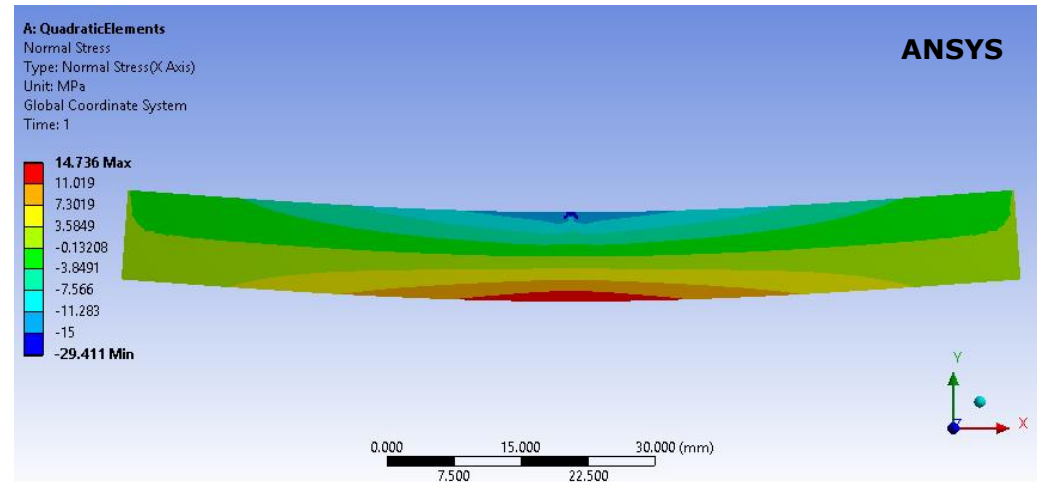
What do you leave OUT when adding to the curriculum

Solve analytically

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 w}{dx^2} \right) = q$$

with 7 different boundary conditions

AND
OR



Two synergetic concepts can typically NOT be learnt in half the time of learning them separately

Uddannelsesmæssig tilgang

The million dollar question:
Hvor generisk vil man gribe det an

