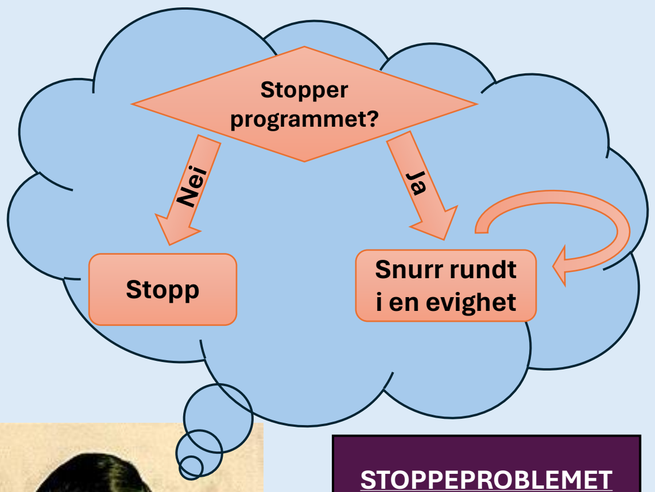
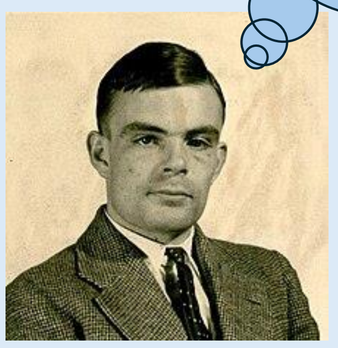
Parameteriserte algoritmer: Løser vanskelige problemer raskt

Høgskulen på Vestlandet

UMULIG ELLER VANSKELIG?





STOPPEPROBLEMET
Q: Kan vi lage et program som sjekker om et annet program stopper opp?
A: Nei!

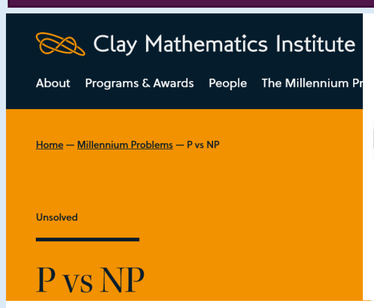
<- Bevis av Alan Turing

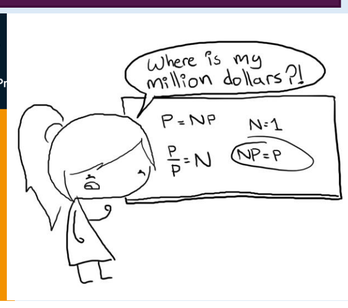
TRE TYPER PROBLEMER

P: Lett å finne en løsning.

NP: Lett å sjekke at en løsning stemmer.

NP-KOMPLETT: Lett å løse bare dersom P = NP.

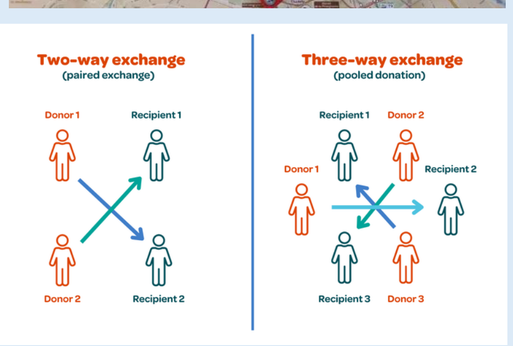




Following the decision of the Scientific Advisory Board, the Board of Directors of CMI designated a $7 million prize fund for the solutions to these problems, with $1 million allocated to the solution of each problem.

$1 million

NOEN NP-KOMPLETTE PROBLEM



ER DETTE NYTTIG? JA!

Her kan du miste jobben...

Dette klarer du (sannsynligvis) ikke...

Men dette kan du få til!

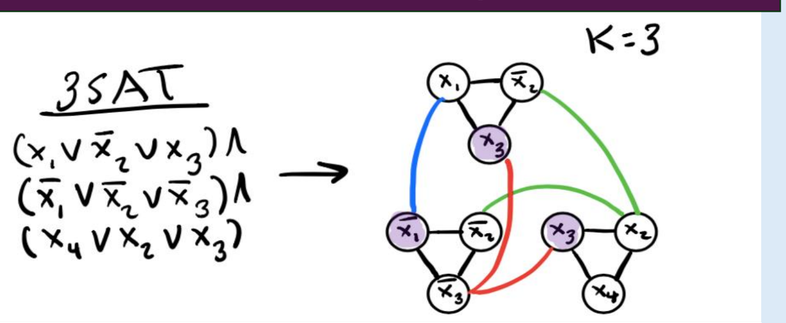
"I can't find an efficient algorithm, I guess I'm just too dumb."

"I can't find an efficient algorithm, because no such algorithm is possible!"

"I can't find an efficient algorithm, but neither can all these famous people."

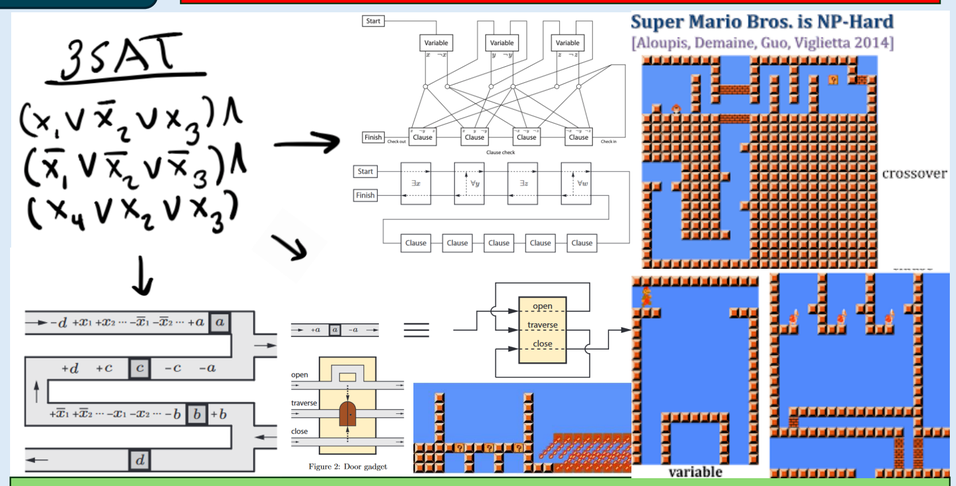
HVORDAN BEVISE A NOE ER NP-KOMPLETT?

IDE:
Vi vet at problemet «3SAT» er NP-komplett.
Dersom vi koder 3SAT som et annet problem er vi i mål!



Noen ganger er dette (ganske) enkelt...

Andre ganger er det mye arbeid...

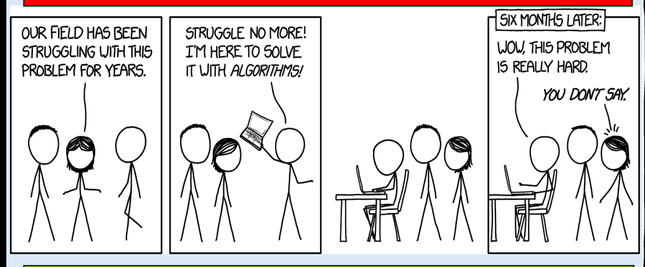


Ofte er jobben alt gjort av noen andre!

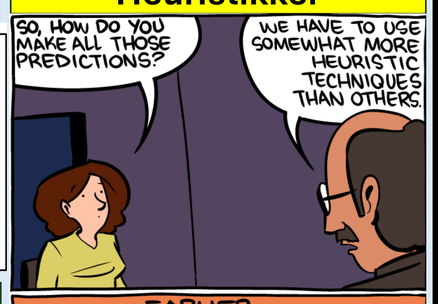
VEIEN VIDERE ->

HVORDAN LØSE VANSKELIGE PROBLEM?

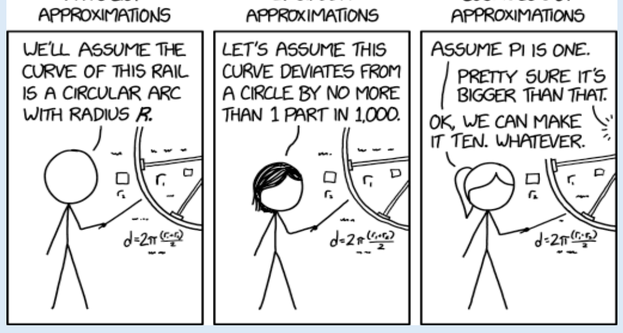
Ingen quick fix, men vi kan bruke...



Heuristikker



Tilnærmingsalgoritmer



Restriksjoner

Randomisering

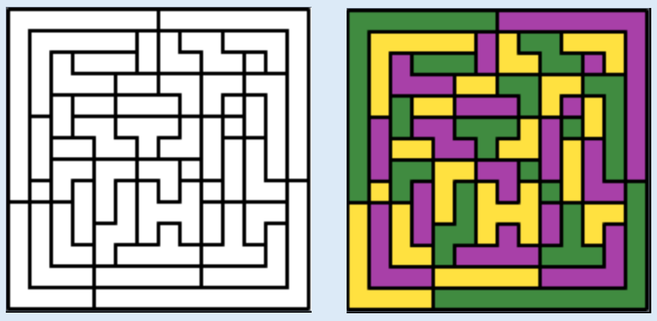
```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

ELLER...

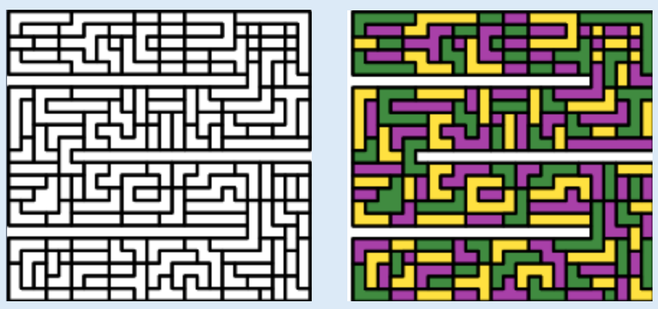
PARAMETERISERTE ALGORITMER!

IDE: Å utnytte «strukturer» i problemet til å finne løsninger som er effektive i praksis.

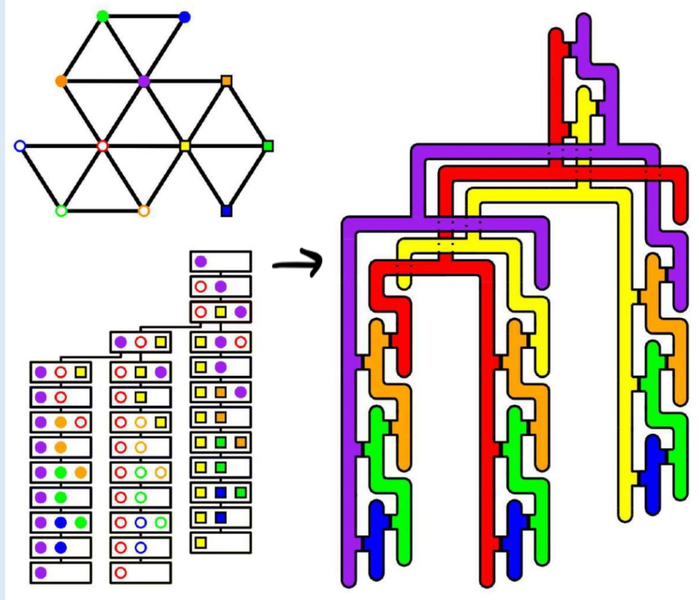
Vanskelig



Enkelt



Parameterisert kompleksitetsteori
For å «bevise» at parameteriserte algoritmer ikke kan hjelpe oss.



Erlend Raa Vågset

expo Høgskulen på Vestlandet