



WIKBORG REIN

Opphavsrettslig beskyttelse av algoritmer

Behovet for beskyttelse

- › Omgir oss hver dag og har påvirkningskraft på avgjørelser vi tar
- › Tilgang til gode og effektive algoritmer er viktig for å skaffe seg sterke markedsposisjoner
 - › Essensielle verktøy for Facebook, Google, Spotify mv. som analyserer brukerdata
 - › Kan rette markedsføring og tjenestene etter brukernes behov

Hvordan kan vi på best mulig måte beskytte verdiene som ligger i algoritmer?

- › Rettslige og ikke-rettslige beskyttelsesmekanismer
- › Dagens tema: Kan de beskyttes som åndsverk etter åndsverkloven § 2?
 - › Men husk: ikke det eneste alternativet!

Den alminnelige oppfatningen

- › Algoritmer er datamaskinprogrammernes ubeskyttede ideer
- › Grunnleggende opphavsrettslig prinsipp: **ideer vernes ikke**
- › Eks: TRIPS art. 9 (2): «Copyright protection extends to expressions and not to ideas, procedures methods of operation and mathematical concepts as such»
- › **Opgavens konklusjon: oppfatningen er misvisende og unyansert**

Hvorfor? Den bygger på en forutsetning om at algoritmebegrepet alltid kan anvendes synonymt med ideer

Programdirektivets fortale pkt. 11:

“(…) only the expression of a computer program is protected and that ideas and principles which underlie any element of a program, including those which underlie its interfaces, are not protected (…). In accordance with this principle of copyright, **to the extent that logic, algorithms and programming language comprise ideas and principles, those ideas and principles are not protected** under this Directive”

Mitt mål? Å problematisere oppfatningen og å vise hvordan algoritmer kan beskyttes etter åvl. § 2

Definisjonsproblematikken: hva er algoritmer?

- › Eksempler:

- › Juridisk litteratur:

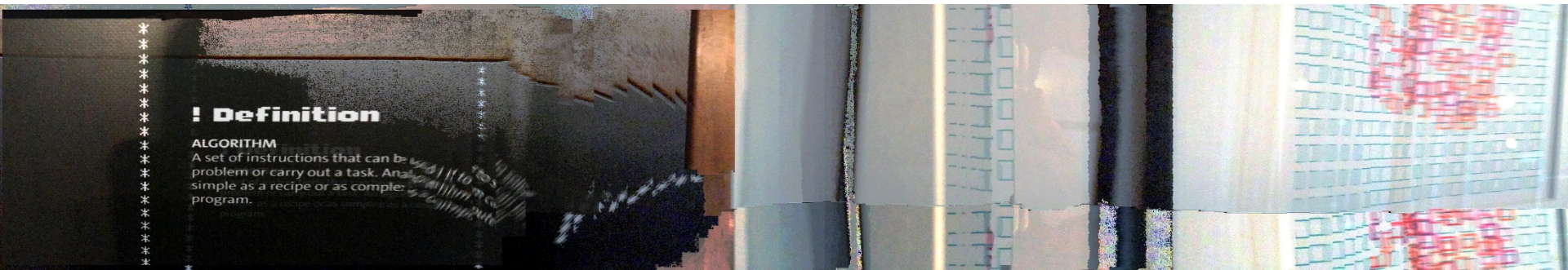
- Programmets generelle princip eller struktur, forstået som den sammenhengende, men trinvisse principielle fremgangsmåte til løsning af et problem* (Koktvedgaard 2005 s. 78)

- › Definisjoner rettet mot informatikk og matematikk:

- Any well-defined computational procedure that takes som value, or set of values, as input and produces som value, or set of values, as output* (Cormen m.fl. 2009)

- › Generelle, overordnede definisjoner:

- A list of instructions for solving a problem* (Cambridge Dictionary)





Oppgavens (vide) definisjon

Et sett med instruksjoner, som trinnvis angir hvordan man skal komme frem til et resultat

- > Hva kan da være algoritmer?
 - > Matteoppgaver, tekniske fremgangsmåter, matoppskrifter, bruksanvisninger til tv-apparater...

Algoritmenes kjennetegn

- › Formålet: produsere resultater (handling, valg/avgjørelser mv.)
- › Instruksjonene skal være så presise at det ikke kreves menneskelig innsats
- › Algoritmen må utføres av en "processor" (menneske, maskin...)
 - › Må derfor være nedskrevet på et språk processoren kan forstå

Undergruppe: de tekniske algoritmene

- > **Tekniske algoritmer:** instruksjoner skrevet på et språk en maskin kan lese og utføre
- > **Maskinlæringsalgoritmer (AI):** en maskins «ability to learn without being explicitly programmed»
 - > «Tradisjonelle» algoritmer: angir spesifikt og presist instruksjonene som må utføres
 - > Maskinlæringsalgoritmer: utvikler og forbedrer fremgangsmåten – instruksjonene forandrer seg ettersom algoritmen gjennomgår en treningsprosess der algoritmen tilegner seg et «minne» som den anvender når den løser ulike oppgaver
 - > Særlig problem: det er vanskelig å identifisere hva som skal beskyttes ettersom algoritmen forandrer seg hele tiden



Åndsverkloven § 2 – vilkårene for opphavsrett

Med åndsverk forstås i denne loven litterære eller kunstneriske verk av enhver art, som er uttrykk for original og individuell skapende åndsinnsett, slik som (...)

§ 2. ¹ Opphavsrett til åndsverk

Den som skaper et åndsverk, har opphavsrett² til verket, og betegnes som opphaver.³

Med åndsverk forstås i denne loven litterære eller kunstneriske verk av enhver art,⁴ som er uttrykk for original og individuell skapende åndsinnsett, slik som

- a) tekster av alle slag, blant annet av skjønnlitterær og faglitterær art
- b) muntlige foredrag
- c) sceneverk, både dramatiske, musikkdramatiske og koreografiske verk, pantomimer og hørespill
- d) musikkverk, med eller uten tekst
- e) filmverk
- f) fotografiske verk⁵
- g) malerier, tegninger, grafikk og lignende billedkunst
- h) skulptur av alle slag
- i) bygningskunst, både tegninger og modeller, og selve byggverket
- j) billedvev og gjenstander av kunsthåndverk og kunstindustri, både forbildet og selve verket
- k) kart, samt tegninger og grafiske og plastiske avbildninger av vitenskapelig eller teknisk art
- l) datamaskinprogrammer⁶
- m) oversettelser og andre bearbeidelser⁷ av verk som er nevnt foran.

For fotografiske bilder som ikke er åndsverk gjelder § 23.

¹ Se kap. 9 jf. res. 21 des 2001 nr. 1563 § 6-1 til § 6-8. – Jf. EØS-avtalen vedlegg XVII nr. 5 (dir 2009/24/EF (programvaredirektivet) art. 1, 2 og 3), nr. 9a (dir 96/9/EF (databasedirektivet) art. 1, 3 og 4), nr. 9f (dir 2006/29/EF (vernetidsdirektivet) art. 6) og nr. 9g (dir 2006/115/EF (utleiedirektivet) art. 2(2)).

² Se § 3.

³ Jf. § 6 (1), § 7 og § 8 (1).

⁴ Se § 14 og § 15.

⁵ Se § 23 (2) annet punktum.

⁶ Se EØS-avtalen vedlegg XVII nr. 5 (dir 2009/24/EF (programvaredirektivet) art. 1).

⁷ Se § 6 (1).

Algoritmens beskyttede uttrykk

- › Ikke fikseringskrav i norsk (eller europeisk) rett
 - › Fikseringskrav: krav til at verket må ha realisert seg i en *fysisk* gjenstand

- › Uten fikseringskrav kan algoritmer tenkes beskyttet på to abstraksjonsnivåer:
 1. Som *beskrivelser* av algoritmen

 2. Som *sammenstillingen av instruksjoner* algoritmen består av



Algoritmer beskyttet som beskrivelser

- Må nedskrives for at prosessoren skal forstå
 - Dette er den litterære *beskrivelsen av algoritmen* som kan beskyttes så lenge den er original
- Beskyttelsen av datamaskinprogrammer
 - Beskyttet etter åvl. § 2 (2) l og programdirektivet
 - Definert som «**et sett med instruksjoner skrevet i programmeringsspråk**», som gjør det mulig «å styre en datamaskin slik at den utfører bestemte oppgaver» (Ot. Prp. Nr. 33)
 - Og vi minner om algoritmedefinisjonen: **et sett med instruksjoner, som trinnvis angir hvordan man skal komme frem til et resultat**

```
bubble_sort.py x
1 # Funksjonen bubble_sort som tar imot en liste med tall (array)
2 def bubble_sort(array):
3     lengthOfArray = len(array) - 1
4
5     # Går gjennom listen
6     for i in range(lengthOfArray):
7         for j in range(lengthOfArray - i):
8             # Sammenligner element j med element j+1
9             # Er element j mindre enn j+1, så bytter de plass
10            if array[j] < array[j + 1]:
11                array[j], array[j + 1] = array[j + 1], array[j]
12
13        return array
14
15
16 if __name__ == "__main__":
17     # Listen med tall som skal sorteres
18     array = [2, 1, 5, 4, 3]
19
20     # Sorterer og skriver ut den nye listen til skjermen
21     print(bubble_sort(array))
22
```

Illustrasjon: KILDEKODE
Odd G. Fatland

```
Algorithm  <algorithm name>
INPUT:  <input specification>
OUTPUT: <output specification>
<statement>;
<statement>;
:
<statement>;
end;
```

Illustrasjon: MATEMATISK ALGORITME
Soicher/Vivaldi 2004

Programvernet beskytter kildekode og objektkode (source or object code)

Algorithm 3.1: MERGESORT(n, X)

comment: Sort the array X of length n

if $n = 2$

then $\left\{ \begin{array}{l} \text{if } X[0] > X[1] \\ \text{then } \left\{ \begin{array}{l} T \leftarrow X[0] \\ X[0] \leftarrow X[1] \\ X[1] \leftarrow T \end{array} \right. \end{array} \right.$

else if $n > 2$

$m \leftarrow \lfloor n/2 \rfloor$

for $i \leftarrow 0$ **to** $m - 1$

do $A[i] \leftarrow X[i]$

for $i \leftarrow m$ **to** $n - 1$

do $B[i] \leftarrow X[i]$

comment: Now sort the subarrays A and B

 MERGESORT(m, A)

 MERGESORT($n - m, B$)

then

$i \leftarrow 0$

$j \leftarrow 0$

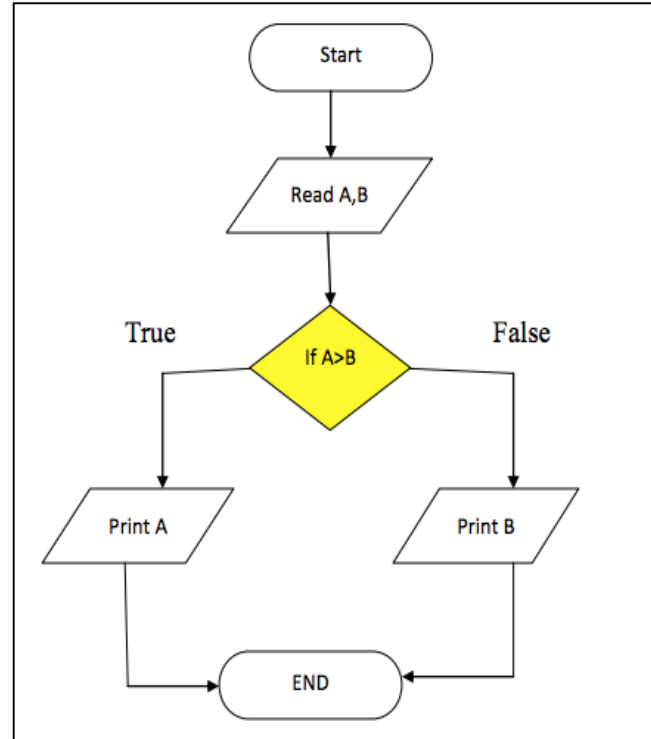
for $k \leftarrow 0$ **to** $n - 1$

if $A[i] \leq B[j]$

then $\left\{ \begin{array}{l} X[k] \leftarrow A[i] \\ i \leftarrow i + 1 \end{array} \right.$ (i)

else $\left\{ \begin{array}{l} X[k] \leftarrow B[j] \\ j \leftarrow j + 1 \end{array} \right.$ (ii)

Illustrasjon: PSEUDOKODE
 Kreher, 2000



Illustrasjon: FLOWCHART
 Mahdi, 2013

Poenget er...

Kildekode, objektkode og forberedende designmateriale er *beskrivelser av tekniske algoritmer* som beskyttes etter åvl. § 2 (2) 1 og programdirektivet så lenge de er originale

Beskrivelser av algoritmer og originalitet

- › Kan beskyttes også utenfor programvernet så lenge de er originale
 - › Andre eksempler fra opphavsretten: faglitteratur, tekniske tegninger og kart
 - › Beskyttes uavhengig av om innholdet beskyttes, jf. Rt. 1997 s. 199 (Cirrus) s. 219 (tekniske tegninger)
- › Opphaver må utøve en kreativ og skapende innsats som gir beskrivelsen et originalt uttrykk
 - › Spørsmålet om opphaver har tatt frie og kreative valg ved beskrivelsen
 - › Valg mellom språk, tegn, illustrasjoner, koder o.l.
 - › Opphaver kan eks. foreta pedagogiske valg ved fremstillingsmåten
- › Definisjonsproblematikk:
 - › Vi bruker begrepene kildekode, objektkode og forberedende designmateriale
 - › Utenfor programvernet – bruksanvisning, oppskrifter o.l.?

Kan sammenstillingen av instruksjoner vernes?

To innfallsvinkler:

- › Strekker programvernets beskyttelsesomfang seg også til dette?
- › Kan sammenstillingen beskyttes utenfor programvernet som eget type verk?

```
16 // Layouts spacing constants
17 #define kVerticalSpacing 5
18 #define kReportAvatarMargin 10
19
20 #pragma mark -
21 #pragma mark Initialization methods
22 - (id) initWithImage: (UIImage*) image {
23     if (self = [super init]) {
24         // Initialize the avatar
25         avatarImage = [[UIImage imageNamed:@"avatar.png"] UIImage];
26
27         // Initialize the avatar name
28         avatarName = [[UILabel alloc] initWithFrame:CGRectMake(0, 0, 100, 20)];
29         [avatarName setText:@"Avatar"];
30         [avatarName setFont:[UIFont systemFontOfSize:12]];
31         [avatarName setTextColor:[UIColor blackColor]];
32         [avatarName setBackgroundColor:[UIColor clearColor]];
33         [avatarName setTextColor:[UIColor blackColor]];
34         [avatarName setBackgroundColor:[UIColor clearColor]];
35         [avatarName setAlpha:0.75];
36
37         // Initialize the avatar
38     }
39     return self;
40 }
```

Beskyttelse under programvernet

- › Utgangspunkt:
 - › Åvl. § 3: råderett over verket «uavhengig av på hvilken måte og i hvilken form», samt verket i «endret skikkelse»
 - › Programdirektivet art. 4: råderetten omfatter «reproduction» og «distribution» i enhver form, men også «translation, adaption, arrangement or any other alteration»
- › C-393/09 (BSA): det som beskyttes under programvernet er det som muliggjør «reproduction or subsequent creation» av dataprogrammet
- › Ren (hel eller delvis) kopiering = klar krenkelse
- › Hva med lignende, men ikke identiske, programmer og oversettelser?

Hvor går grensen ?

- › C-393/09 (BSA) avsnitt 35: programdirektivet beskytter «**the expression in any form of a computer program which permits reproduction in different computer languages**, such as the source and the object code»
- › Generaladvokaten: beskyttelsen av dataprogrammer ikke er «confined to the literal elements of that program, that is to say, the source and the object code, **but extends to any other element** expressing the creativity of its author» (Opinion of Advocate General Bot in case C-406/10, 2011)
- › C-406/10 (SAS Institute): ingen krenkelse fordi saksøkte (WPL) ikke hadde hatt tilgang til, og derfor ikke kunne kopiere, kildekoden
 - › Gjengir COM (88) 816 pkt. 3.7: «protection covers only the individual expression of the work and thus leaves other authors the desired latitude to create similar or even identical programs *provided that they refrain from copying*» (44)

Oppsummering underveis...

- › Vi kan konkludere med at tekniske algoritmer beskyttes under programvernet som beskrivelser, og at originale beskrivelser av algoritmer kan tenkes beskyttet også utenfor programvernet
- › Men beskyttelsen er teoretisk interessant og har liten praktisk betydning på grunn av definisjonene vi bruker
 - › Én viktig, praktisk, konsekvens: den alminnelige oppfatningen kan være forvirrende og skape misforståelser
- › Rettspraksis tyder på at sammenstillingen av instruksjoner ikke er beskyttet under programvernet slik rettstilstanden er i dag

Beskyttelse av sammenstillingen – utenfor programvernet

- › Påminnelse: begrenser seg ikke til tekniske algoritmer, men har størst verdi og betydning på dette området
- › Vilkåret: sammenstillingen må være original i opphavsrettslig forstand
 - › Originalitetskravet lovfestet i 2018 i åvl. § 2 (2): verket må være "uttrykk for original og individuell skapende åndsinnset"
 - › EU-domstolen: verket må være "the author's own intellectual creation" – C-5/08 (Infopaq I)
 - › Vektlagt valgmuligheter: om opphaver har utøvet frie og kreative valg



Algoritmenes funksjonelle preg

- › Oppgaven er å produsere resultater
- › Grunnleggende: originalitetskravet trekker en grense mot det rent funksjonelle
 - › Kan medføre sterk begrensning i utviklingen og beskytte nødvendige elementer for funksjonelle formål
 - › Ved originalitetsvurderingen begrenser de funksjonelle elementene naturlig opphavers valgmuligheter
- › Må utelukkes fra vurderingen:
 - › Algoritmens nytteverdi/funksjon: hvilket problem den løser
 - › Hva/hvem som er algoritmens prosessor
- › Parallell fra programvernet: det beskyttede er det som *muliggjør reproduksjon av algoritmen*, herunder *sammenstillingen av instruksjonene*



Hva som beskyttes av opphavsretten – vernet av sammenstillinger

- › Originalitet er ikke lik nyhet
 - › Kan bygge på kjente elementer som er "sammenstilt på en slik måte at verket som helhet fremstår som originalt" (Rt. 2007 s. 1329 – Huldra i Kjosfossen, 44)
- › Alle verk er til en viss grad bare sammenstillinger
 - › Eks. sammenstillingen av noter og akkorder for å lage musikkverk
- › Bernkonvensjonen art. 2 (5) – vernet av "kompilasjoner"
 - › Eks. kokebøker, telefonkataloger, diktsamlinger, databaser osv.
 - › Originaliteten ligger i bl.a. utvelgelsen, struktureringen, sorteringen, planleggingen osv.
- › Kan det å lage algoritmer være original, skapende åndsinnsetts ved utvelgelsen, struktureringen, sorteringen, planleggingen og sammenstillingen av instruksjonene den består av?



Å lage algoritmer

- > 3 steg: sekvensiering, seleksjon og iterasjon
- > Begreper fra teknologien
- > Algoritmen for å steke et speilegg
- > A – Hell egget over i stekepannen
- > B – Knekk egget i to
- > C – Varm opp stekepannen
- > **Sekvensieringen:** valg av rekkefølge – C-B-A eller B-A-C?
- > **Seleksjon:** vi innfører et alternativ algoritmen kan velge basert på kriterier
- > D – Dersom stekepannen er av jern, tilfør smør
- > **Iterasjon:** vi lar algoritmen gjenta hele eller deler av seg selv
- > Gjenta A og B: algoritmen steker 2 egg
- > Algoritmen blir en byggekloss
- > Kan kombineres med andre

Uttrykker dette originalitet?

- › Morten Rosenmeier: problematisk å anse algoritmer som ideer der datamaskinprogrammene er sammensatt av mange små "basialgoritmer"
- › Utviklerens valg mellom flere tusen, mindre algoritmer danner ofte grunnlag for store, komplekse algoritmer – bør dette vernes?
 - › EU-kommisjonen mente datamaskinprogrammer burde vernes fordi de utgjør "the selection and compilation" av programmets underliggende ideer og logikk (COM (88) 816 s. 5-6)
 - › Er også de store, komplekse sammenstillingene bare uttrykk for ideer og logikk?

Hva er ideer?

> TRIPS-avtalen art. 9 (2) likestiller ideer med prosedyrer, metoder og prosesser

> Få europeiske rettskilder tar for seg hva dette er

> Ordbøker til hjelp?

> Metode er en "planmessig fremgangsmåte (...) gjerne grunnet på regler og prinsipper"

Store Norske Leksikon

> Prosedyre er "a set of actions that is the official or accepted way of doing something"

Cambridge Dictionary

> Prosess er "a series of actions that you take in order to achieve a result"

Cambridge Dictionary



Er algoritmer alltid "ideer"?

- > Innebærer dette at alt som kan klassifiseres som "fremgangsmåter" er ubeskyttet
- > Bare der de er styrt av etablerte regler og prinsipper?
- > Hvorfor ikke beskytte ideer (metoder, prosedyrer, prosesser...)?
 - > Vi ønsker utvikling og innovasjon
 - > Nødvendige steg i en fremgangsmåte bør ikke begrenses
 - > Men kanskje bare der det ikke finnes tilstrekkelige valgmuligheter for å komme frem til samme resultat?
- > Jo mer komplekse sammensetningene av instruksjoner er – jo større valgfrihet vil det som regel være

i d e a s



Problemer ved en eventuell beskyttelse

- › Algoritmene bygger på logiske slutninger (ofte matematiske beregninger)
 - › Begrenser valgfriheten
 - › Fremgangsmåten er alltid styrt av en eller annen form for logikk
- › EU-domstolen synes å begrense til kun konkrete beskrivelser
 - › Kan tale for at sammenstillingene er ment å unntas i alle tilfeller
- › Vilkåret om at åndsverket må være litterært eller kunstnerisk
 - › Målet med sammensetningen er å raskest og på best mulig måte nå et mål

Rettspolitisk perspektiv



- › Utviklingen av komplekse algoritmer er kommet utrolig langt
 - › Bør kanskje beskyttes på et mer abstrakt nivå?
- › Ofte mange veier til samme mål – og dermed mange valgmuligheter
 - › Målet er å finne de beste veiene for å hevde seg i konkurransen med andre
- › Rettspraksis er ikke entydig – konklusjonen er ikke åpenbar

Er vernet hensiktsmessig?

- › Så lenge kun *beskrivelsene* vernes er det neppe tilstrekkelig
 - › Hvis vi kun verner svært nærliggende kopier og etterligninger
 - › PS: vern av beskrivelser er ikke uten verdi – programvernet er et godt eksempel

- › Er det hensiktsmessig å verne sammenstillingene av instruksjoner?
 - › Definisjonsproblemene: må ha en felles forståelse av hva det er, og de vernede elementene
 - › Høyst sannsynlig u håndterlig og vanskelige originalitetsvurderinger i praksis
 - › Skaper lite forutsigbarhet
 - › Henger sammen med opphavsretten som en *naturlig* rettighet
 - › Sammenlign patentretten: oppfinnelsene må registreres og offentliggjøres



Oppsummering

- › Målet: problematisere den alminnelige oppfatningen

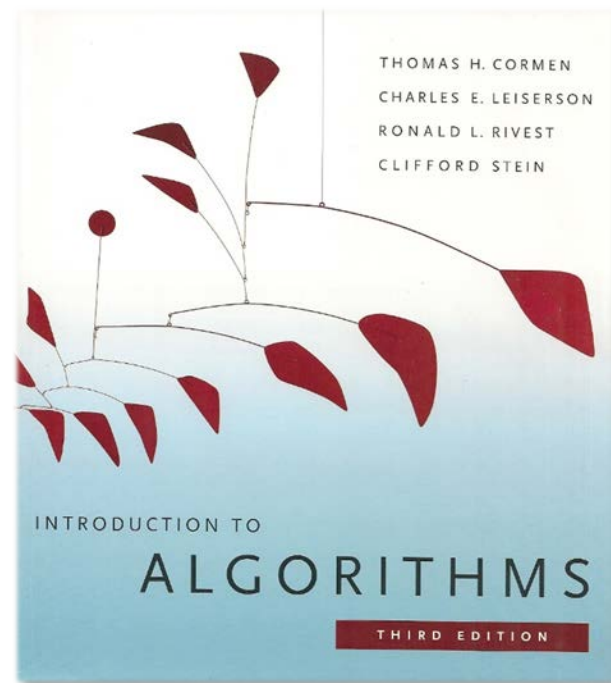
- › Mener den er unyansert og kan skape misforståelser

Mitt råd:

- › Vær oppmerksom på definisjonsproblematikken og utvis tilbakeholdenhet med å kommunisere det til klienter

- › Konklusjonene er:

- › Beskyttes i en viss utstrekning etter åvl. § 2 som originale *beskrivelser*
 - › Neppe tilstrekkelig
- › Gode grunner taler for at sammenstillingene av instruksjoner må anses som ubeskyttede ideer, men dette bør heller ikke avskrives
 - › Høyst sannsynlig uhåndterlig i praksis og lite forutberegnelig



Andre alternativer?

- › Patentretten mer forutberegnelig og kanskje mer naturlig, immaterialrettslig vern?
- › Forretningshemmeligheter i mfl. § 28?
- › I strid med "god forretningskikk" i mfl. § 25 eller etterligningsvernet i mfl. § 30 å bruke konkurrentenes algoritmer?
- › Indirekte beskyttelse gjennom vernet av tekniske beskyttelsesmekanismer i åvl. § 99
- › Aktuelt med et "sui generis"-vern?
 - › Flytte vurderingen fra den skapende, kreative innsatsen i verksvurderingen, til hvilke ressurskrevende investeringer som er foretatt

Maskinlæringsalgoritmer

- › En maskins "ability to learn without being explicitly programmed"
- › Prosessen ved utvikling av maskinlæring:
 - › Utvikler innhenter store mengder data fra relevante kilder som prosesseres og transformeres
 - › Algoritmen trenes opp ved bruk av ulike maskinlæringsmetoder
 - › Eks. supervised learning (bruk av klassifisert data) vs. reinforcement learning (læring ved prøving og feiling)
 - › Prosessen justeres og evalueres til man får ønskede resultater



Identifiseringsproblemet

- › Opphavsrettslige utfordringer:
 - › Kan vi beskytte algoritmer som stadig forandrer seg?
 - › Kan vi identifisere og objektivt beskrive en maskinlæringsalgoritme?

- › C-310/17 (Levola Hengelo) avsnitt 40:
 - › Uttrykket kan beskyttes "even though that expression is not necessarily in a permanent form"
 - › Men må være "identifiable with sufficient precision and objectivity"



Foto: Wikborg Rein, Erik Burås/STUDIO B13, Ilya Hendel, Helge Hansen/Montag, istockphoto.com

Ansvarsforhold: Denne presentasjonen inneholder en overordnet beskrivelse av enkelte regler i norsk rett. Den utgjør ikke juridisk rådgivning, og ingen forretningsmessige beslutninger bør baseres på den.