



WIKBORG REIN

Opphavsrettslig og annen beskyttelse av algoritmer

Definisjonsproblematikken: hva er algoritmer?

- › Eksempler:

- › Juridisk litteratur:

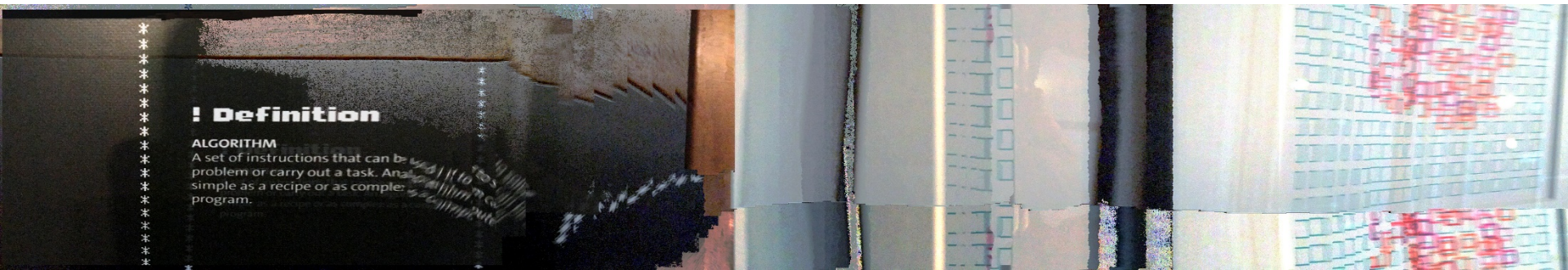
- Programmets generelle princip eller struktur, forstået som den sammenhengende, men trinvise principielle fremgangsmåde til løsning af et problem (Koktvedgaard 2005 s. 78)*

- › 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)*





(Vid) 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...
- › Kjennetegn
 - › Formålet: produsere resultater (handling, valg/avgjørelser mv.)
 - › Instruksjonene skal være så presise at det ikke kreves annen menneskelig innsats enn å følge dem
 - › Må utføres av en "prosessor" (menneske, maskin...)

Undergruppe: de tekniske algoritmene

- › **Tekniske algoritmer:** instruksjoner skrevet på et språk en maskin kan lese og utføre
- › **Maskinlæringsalgoritmer (AI):** maskinlæring er en maskins «*ability to learn without being explicitly programmed*»
 - › «Tradisjonelle» algoritmer: angir spesifikt og presist instruksjonene som må utføres
 - › Maskinlæringsalgoritmer: utvikler og forbedrer kontinuerlig fremgangsmåten
 - › Særlig problem: det er vanskelig å identifisere hva som skal beskyttes ettersom algoritmen forandrer seg hele tiden



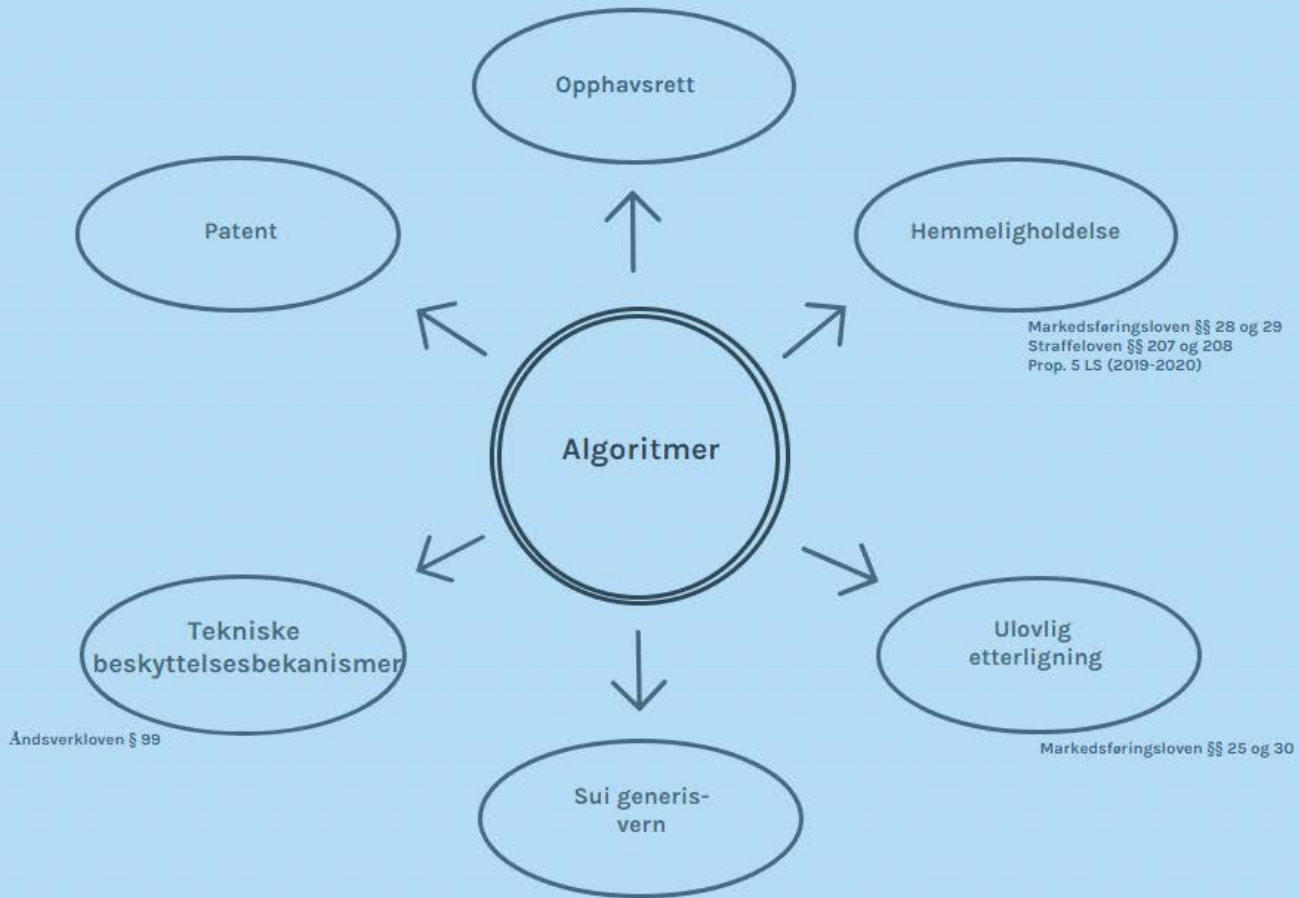
Behovet for beskyttelse

- › 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 bedrifter som analyserer brukerdata
 - › Kan rette markedsføring og tjenestene etter brukernes behov

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

- › Dagens hovedtema: Kan de beskyttes som åndsverk etter åndsverkloven § 2?
 - › Men husk: ikke det eneste alternativet!





Opphavsretten: 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»

› Oppgavens 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

§ 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).



Hvilke uttrykk kan beskyttes?

Kan tenkes beskyttet på to abstraksjonsnivåer:

1. Som *beskrivelser* av algoritmen
2. Som *sammenstillingen av instruksjoner og flere små algoritmer*



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) 1 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

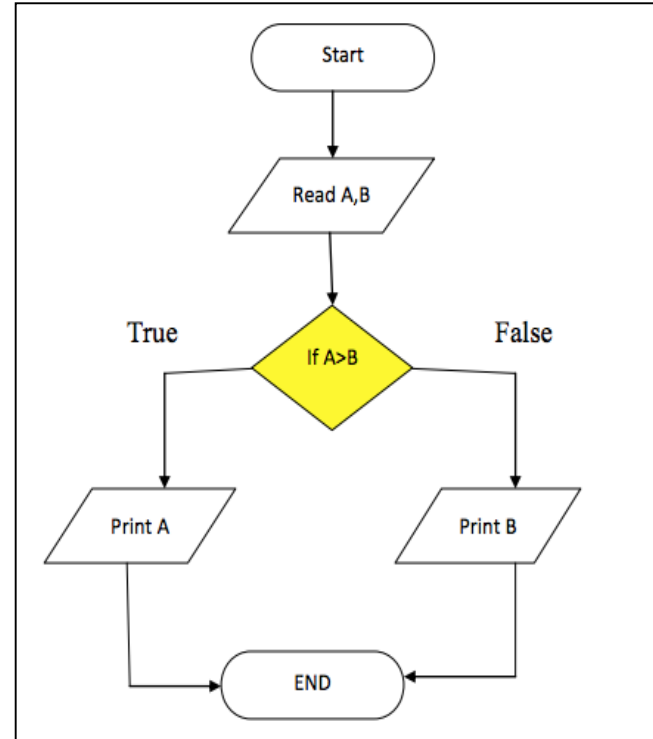
Programvernet beskytter kildekode og objektkode (source or object code)

```

Algorithm 3.1: MERGESORT( $n, X$ )
comment: Sort the array  $X$  of length  $n$ 
if  $n = 2$ 
  then {
    if  $X[0] > X[1]$ 
      then {
         $T \leftarrow X[0]$ 
         $X[0] \leftarrow X[1]$ 
         $X[1] \leftarrow T$ 
      }
    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 {
               $X[k] \leftarrow A[i]$ 
               $i \leftarrow i + 1$ 
            } (i)
            else {
               $X[k] \leftarrow B[j]$ 
               $j \leftarrow j + 1$ 
            } (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
 - › Beskyttes uavhengig av om innholdet beskyttes, jf. Rt. 1997 s. 199 (Cirrus) s. 219 (tekniske tegninger)
- › Opphaver må utøve en kreativ og skapende innsats
 - › Frie og kreative valg ved beskrivelsen?
 - › Valg mellom språk, tegn, illustrasjoner, koder, pedagogiske valg o.l.
- › Definisjonsproblematikk:
 - › Vi bruker begrepene kildekode, objektkode og forberedende designmateriale
 - › Utenfor programvernet – bruksanvisning, oppskrifter o.l.?

Algoritmer som sammenstillingsverk

Er det å lage algoritmer uttrykk for original, skapende åndsinnset, jf. åvl. § 2 (2)?

- › "(t)he author's own intellectual creation" – C-5/08 (Infopaq I)
 - › EU-domstolen vektlegger valgmuligheter: om opphaver har utøvet frie og kreative valg
- › Grense mot det funksjonelle
 - › Oppgaven er å produsere resultater
 - › Må utelukkes fra vurderingen:
 - › Algoritmens nytteverdi/funksjon: hvilket problem den løser
 - › Hva/hvem som er algoritmens prosessor

```
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 = [[UIImageView alloc] initWithImage:image];
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 }
```



Opphavsrettens vern 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

- › 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**
- > **Sekvensiering:** 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

Veiledning fra rettspraksis? - 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)
 - › Hva ligger i kopieringsbegrepet?

Originalitet – eller bare ideer?

- › Morten Rosenmeier: problematisk å anse algoritmer som ideer der datamaskinprogrammene er sammensatt av mange små "basialgoritmer"
 - › Sammenstillinger av ubeskyttede ideer kan beskyttes av opphavsretten

i d e a s

- › Utviklerens valg mellom flere tusen, mindre algoritmer danner ofte grunnlag for store, komplekse algoritmer – kan dette vernes?
 - › Styr av logikk og nødvendighet
 - › Målet med sammensetningen er å raskest og på best mulig måte nå et mål – ikke skape noe litterært eller kunstnerisk
 - › Men hva hvis det finnes tilstrekkelige valgmuligheter for å komme frem til samme resultat?

Rettspolitisk perspektiv – begrunnelse for vern



- › Utviklingen har kommet langt
- › 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
 - › Jo mer komplekse sammensetninger – jo større valgfrihet
- › Rettspraksis er ikke entydig – konklusjonen er ikke åpenbar



Er et sammenstillingsvern hensiktsmessig?

- › Definisjonsproblemerne: må ha en felles forståelse av hva som vernes
- › Høyst sannsynlig uhåndterlig og vanskelige originalitetsvurderinger i praksis
- › Lite forutsigbarhet
 - › Henger sammen med opphavsretten som en *naturlig* rettighet
- › Lang vernetid (70 år)
 - › Vi ønsker teknologisk utvikling og innovasjon
 - › Jussen vil ikke henge med i utviklingen

Andre muligheter?

- › Patentretten
 - › Beskyttelse av *fremgangsmåter*, mer *forutberegnelig* og *tidsmessig mer begrenset*
- › Hemmeligholdelse
 - › Algoritmen som "forretningshemmelighet" – må være *hemmelig*, ha *kommerseil verdi* og være hemmeligholdt ved *rimelige tiltak* jf. forslag til ny lov om forretningshemmeligheter § 3
 - › Ingen enerett til algoritmen som sådan
- › Markedsføringsloven §§ 25 og 30
 - › Mindre forutsigbarhet, men utvei i konkrete tilfeller
 - › Kombinasjon med hemmeligholdelse
- › Tekniske beskyttelsesmekanismer, åvl. § 99
 - › Gir kun indirekte beskyttelse
- › Aktuelt med et "sui generis"-vern?
 - › Flytte vurderingen fra den skapende, kreative innsatsen i verksvurderingen, til hvilke ressurskrevende investeringer som er foretatt
 - › Kanskje fornuftig, men krever lovarbeid, faste rammer mv.

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 si at algoritmer aldri kan beskyttes

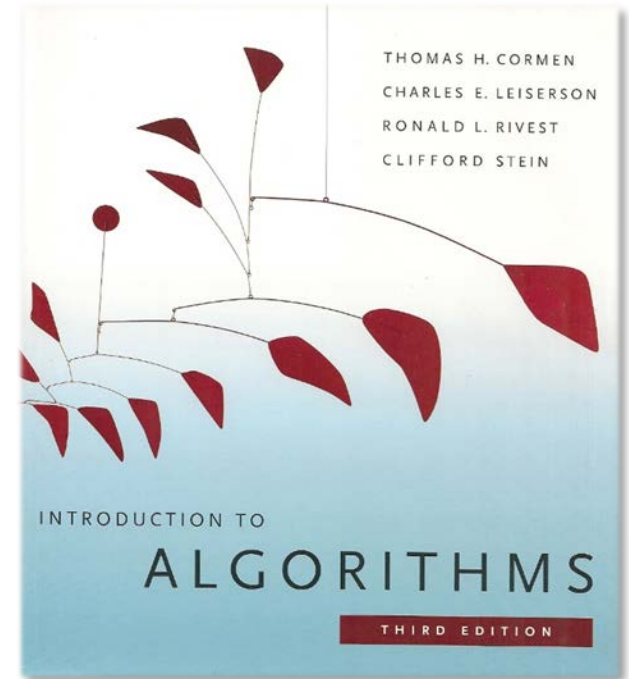




Foto: Wikborg Rein, Erik Burås/STUDIO B13, Ilya Hendel, Helge Hansen/Montag, istockphoto.com

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.