

# Pensum i INF 4130/9135

---

I år var det dessverre store problemer med tilgjengelighet av hovedlæreboken (Berman & Paul: *Algorithms: Sequential, Parallel, and Distributed*), og dette endte med at vi trykket et kompendium med stoff fra denne. Forlagsavtaler gjorde imidlertid at ikke alt pensumstoff fikk plass i kompendiet. Av pensum fra hovedlæreboken er det bare kapitlene 8, 10, 14 og 23 som fikk plass, kapitlene 9 og 20 må skrives ut fra kurssidene. I tillegg inneholder kompendiet også kapittel 26 om *NP*-komplekthet. Kapitlet er nok litt detaljert, men kan være nyttig som alternativ tilnærming, eller hjelpestoff, til pensumet fra Karabeg & Djurhuus: *Kompendium 51 til IN210*. Dessuten er det angitt noe hjelpestoff om triangulering tatt fra et kompendium fra et annet universitet (se detaljer lenger ned).

Ved siden av det som er angitt under, forventer vi at oppgavestoffet, både fra ukeoppgavene og de obligatoriske oppgavene, er forstått. Tilsvarende forhold skal man også ha til foilene fra forelesningene, men noen steder er det også angitt foilene eksplisitt er pensum.

**Som tidligere år: Alle trykte og skrevne hjelpemidler er tillatt på eksamen.**

## **Fra Berman & Paul: *Algorithms: Sequential, Parallel, and Distributed***

(Husk å studere trykkfeillista. Merk at noen feil er rettet i utgaven som ble trykket opp):

Kapittel 8 (Divide-and-conquer):

- Innledningen av kapitlet og kap. 8.1 gir litt bakgrunn for top-down-rekursjon og memoisering
- Kap 8.6.2 er pensum

Kapittel 9 (Dynamisk programmering):

- Hele

Kapittel 10: (Backtracking og branch-and-bound):

- Her summerte vi bare opp enkle metoder for fullstendig søk: Dybde først søk (DFS, *Backtracking*), og Bredde-først-søk (BFS, *Branch-and-bound*) med LIFO-, FIFO- og Prioritets-kø (dette var i stor grad repetisjon fra INF 2220/1020). Hovedprinsippene gjennomgått i kapitlet er pensum, men ikke eksemplene etc.

Kapittel 14 (Matching og flyt i nettverk):

- Alt unntatt 14.1.3 og 14.2.6.
- Det brukes en del formalisme og terminologi i boka som vi ikke brukte på forelesningen (se foilene), og detaljene i dette er ikke pensum

Kapittel 20 (Strengsøk):

- Hele

Kapittel 23 (A\*-søk og spilltrær):

- Alt unntatt 23.4

## **Om triangulering:**

- Her er det foilene om triangulering fra 17/11 som er pensum.
- Som bakgrunn kan man også lese tilsvarende stoff i et utsnitt fra et kompendium. Dette utsnittet kan hentes under egen overskrift på kurssiden. Mye er her tatt med som rent bakgrunnsstoff, men det som nærmest tilsvarer vårt pensum er det angitt under. Merk at valg av hva som er

definisjoner og hva som er teoremer er gjort litt annerledes her enn på foilene:

- Fra starten av Lecture 17, til (men ikke med) "Closest pair property" på side 75.
  - Side 76, fra "Maximizing Angles and Edge Flipping" fram til avsnittet "We will not give ..." nederst på siden.
  - Fra starten av Lecture 18, og fram til "We will not prove ..." på side 78.
  - Fra "Incremental update" på side 78 fram til avsnittet "There is only one major ..." nederst side 79.
- Om man vil se på hele kompendiet ligger det på:  
<http://www.cs.umd.edu/~mount/754/Lects/754lects.pdf>

### **Fra M.A.Weiss: *Data Structures and Algorithm Analysis in Java***

(De aktuelle sidene kan lastes ned fra undervisningsplanen, om man ikke har boka fra INF 2220.)

Kapittel 6 (Prioritetskøer):

- Alt unntatt 6.5 og 6.7.

Kapittel 11:

- 11.4 (Fibonacci-heaper): I tillegg til selve algoritmene inngår det her å forstå og kunne de tidsgrensene som er angitt, men bevisene for disse er ikke pensum.

### **Fra Karabeg og Djurhuus: *Kompendium 51 til IN210*:**

[www.uio.no/studier/emner/matnat/ifi/INF4130/h10/undervisningsmateriale/in210\\_komp99.pdf](http://www.uio.no/studier/emner/matnat/ifi/INF4130/h10/undervisningsmateriale/in210_komp99.pdf)

Merk at Forelesning 1 tilsvarer kapittel 3.1 osv.

- Forelesning 1. side 17 – 29
- Forelesning 2. side 37-39, 46-52
- Forelesning 3. side 72-78
- Forelesning 5. side 106-131
- Forelesning 6. side 132-142 + Cooks teorem bevis-idé, 148-159
- Forelesning 7. side 160-177

(På [www.uio.no/studier/emner/matnat/ifi/INF4130/h10/undervisningsmateriale/in210\\_extract.pdf](http://www.uio.no/studier/emner/matnat/ifi/INF4130/h10/undervisningsmateriale/in210_extract.pdf) ligger et utdrag av de aktuelle sidene.)

## **Oversikt**

Gjennom kompleksitetsteorien [Karabeg og Djurhuus: *Kompendium 51 til IN210*] har vi lært at, og hvordan, problemer kan deles inn i medgjørlige ( $P$ ), umedgjørlige ( $NP$ -komplette) og uløselige (*unsolvable*) problemer, og vi har sett litt på alternative teknikker for løsning av umedgjørlige problemer.

Av vanlige algoritmeteknikker har vi sett på dynamisk programmering [Kap. 9 B&P], en metode som både kan brukes både på medgjørlige og umedgjørlige problemer. Strengsøk [Kap. 20 B&P] (bla i bioinformatikk) ledet opp til dette, i denne sammenhengen så vi også på memoisering og *Divide-and-conquer*. *Divide-and-conquer* kom vi også tilbake til i de geometriske algoritmene [Kap. 8.6 B&P].

Mange problemer er umedgjørlige når vi forsøker å danne tripler, fargelegger med tre farger etc., men medgjørlige når vi bare skal danne par, bruker to farger etc. (Det er dog ingen regel at det alltid er et

slikt skille mellom to og tre, men mange ganger er det slik.). Matchingteknikker er ofte nyttige når det er snakk om par, to farger, klausuler av størrelse to, etc. Vi har sett på sammenhengen mellom matching og flyt i grafer, og studert algoritmer for flyt i generelle grafer [Kap. 14 B&P].

Vi har også sett på søk som en generell løsningsmetode [Kap 10, 23 minus 23.4 B&P] (mange problemer kan løses med en eller annen form for søk i et tilstandsrom) og på spilltrær (bla for sjakk). Prioritetskøer er nødvendige i søkealgoritmene [Kap 6 minus 6.5 og 6.7 pluss 11.4 MAW].

[slutt]