

Endelig pensum INF 4130

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.

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 siste utgave):

Kapittel 3 (O-notasjon):

- 3.1.1 – 3.1.5 omhandler *O*-notasjon etc., dette regner vi vel som kjent stoff, stoffet i 3.6 g 3.7 har vi dekket på anen måte, men les det gjerne her også.

Kapittel 8 (Divide-and-conquer):

- Bare 8.6

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 kapittelet er pensum, men ikke eksemplene etc.

Kapittel 14 (Matching og flyt i nettverk):

- Fra boka, alt unntatt 14.1.3 og 14.2.6.
- I forbindelse med matchinger er også foil 11 (ommatchinger i generelle grafer) fra 11/11 pensum.
- I forbindelse med 14.2.7 er også foil 27 (siste) fra 11/11 direkte pensum.
- Dessuten skal man vite (se ukeoppgave 4 fra 16/11) at for bipartite grafer kan man alltid finne en maksimal matching og et minimalt kantoverdekkende nodeutplukk som er like store (men bevisene og andre detaljer rundt dette er ikke direkte pensum.)
- Det brukes en del formalisme og terminologi i dette kapittelet 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

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

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 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]