NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4120 Algoritmer og datastrukturer

Faglig kontakt under eksamen	Magnus Lie Hetland
Telefon	918 51 949

Eksamensdato	15. august, 2018
Eksamenstid (fra–til)	09:00–13:00
Hjelpemiddelkode/tillatte hjelpemidler	D

Målform/språk	Bokmål
Antall sider (uten forside)	2
Antall sider vedlegg	0

Informasjon om trykking av eksamensoppgave

Originalen er

1-sidig ☑ **2-sidig** ☐

sort/hvit ☑ **i farger** ☐

Skal ha flervalgskjema ☐

Kontrollert av

Dato Sign

Merk: Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Les dette nøye

(i) Les hele eksamenssettet nøye *før* du begynner!

(ii) Faglærer går normalt én runde gjennom lokalet. Ha evt. spørsmål klare!

(iii) Skriv bare så mye du mener er nødvendig. Svar som inneholder irrelevante aspekter kan regnes som helgardering, og det kan trekke ned!

Oppgaver

5% 1. Betrakt følgende utsagn (der reduksjonene antas å ta polynomisk tid):

(i) Et problem B i **NP** er **NP**-komplett dersom alle problemer i **NP** kan reduseres til B.

(ii) For $A, B \in$ **NP**, der A er **NP**-komplett: Dersom vi kan redusere fra A til B så er B **NP**-komplett.

Forklar kort hvorfor (ii) følger av (i).

5% 2. Hva er kjøretiden til Insertion-Sort på en sortert tabell?

5% 3. Hva sier heltallsteoremet (*the integrality theorem*)?

5% 4. Om vi viser at grådighetsegenskapen (*the greedy-choice property*) holder har vi likevel ikke vist at grådighet er korrekt. Forklar.

5% 5. Du starter med en sammenhengende graf $G = (V, E)$ og sletter kanter fra E til det finnes nøyaktig én sti mellom ethvert par med noder $u, v \in V$. Hva kalles den resulterende grafen $G' = (V', E)$? (Merk: Her spørres det etter forholdet mellom G' og G, ikke hva G' er isolert.)

5% 6. Du prøver å implementere BFS for urettede grafer, men på grunn av en kodefeil, er rekkefølgen på nodene i køen din ikke lenger FIFO, men helt vilkårlig. Kan du nå være sikker på å besøke alle nodene? Forklar.

5% 7. Vi kan memoisere en funksjon, så vi slipper å regne den ut flere ganger om den kalles med samme argumenter igjen. Det kan være en nyttig optimalisering i praksis, som en form for hurtigminne (*cache*), men har det noen teoretisk betydning for asymptotisk kjøretid? Forklar.

5% 8. Hvordan fungerer Relax(u, v, w)?

5% 9. Hva er forskjellen på konkrete og abstrakte problemer, slik boka definerer dem?

5% 10. Hvordan fungerer List-Delete(L, x)?

5% 11. Hvordan kan vi si at Merge-Sort er optimal? Forklar.

5% 12. Anta at du har funnet den transitive lukningen (*transitive closure*) $G^* = (V, E^*)$ av grafen $G = (V, E)$, i form av en boolsk matrise. Vis hvordan G^* kan oppdateres effektivt dersom en kant $\{u, v\}$ legges til i G. Hva blir kjøretiden?

5% 13. Anta at du starter med et tomt binært søketre og setter inn n verdier fra verdiområdet $1, \dots, k$, der $k \leq n$. Rekkefølgen på verdiene er $1, 2, \dots, k, 1, 2, \dots, k, \dots$, altså verdiene fra 1 til k gjentatte ganger. Hva blir høyden til søketreet? Oppgi svaret som funksjon av n og k. Du kan anta at n er delelig på k. Ikke bruk asymptotisk notasjon. Forklar kort, gjerne med en figur.

(Merk: For å følge bokas implementasjon, skal nodene i venstre deltre være strengt mindre enn rota. Husk at høyden måles i antall kanter, ikke noder.)

5% 14. Løs følgende rekurrens: $\mathrm{T}(n) = 4\mathrm{T}(n/2) + 1/\sqrt{n}$. Skriv svaret med Θ-notasjon.

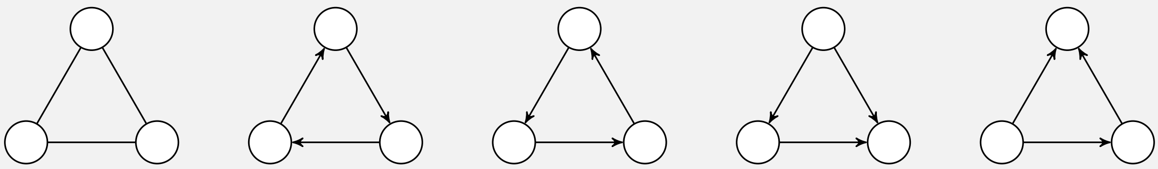
Den såkalte *Ackermannfunksjonen* defineres slik:

$$\mathrm{A}(m,n) = \begin{cases} n+1 & \text{hvis } m = 0; \\ \mathrm{A}(m-1, 1) & \text{hvis } m > 0 \text{ og } n = 0; \\ \mathrm{A}(m-1, \mathrm{A}(m, n-1)) & \text{hvis } m > 0 \text{ og } n > 0; \end{cases} \tag{1}$$

der m og n er ikke-negative heltall.

5% 15. Hva er $\mathrm{A}(1, n)$ som funksjon av n? Ikke bruk asymptotisk notasjon. Vis utregning.

En *orientering* av en urettet graf $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ er en tilordning av retning til hver kant $e \in \mathrm{E}$, som resulterer i en ny rettet graf $\mathrm{D} = (\mathrm{V}, \mathrm{A})$, der hver urettet kant $\{u, v\}$ i E tilsvarer en rettet kant (u, v) eller (v, u) i E, og omvendt. Nedenfor er et eksempel på en urettet graf G og noen orienteringer D_i.



G D_1 D_2 D_3 D_4

Her er D_3 og D_4 eksempler på *asykliske* orienteringer. En *sterk* orientering er *sterkt sammenhengende* (*strongly connected*, det vil si, fra enhver node u finnes det en rettet sti til enhver annen node v). Eksempler på dette er D_1 og D_2. Disse finnes bare hvis ethvert snitt inneholder minst to kanter. I en *k-orientering* har enhver node v maksimalt $k(v)$ inn-kanter, for en funksjon $k : \mathrm{V} \to \mathbb{N}$.

5% 16. Din venn Smartnes prøver å finne én bestemt orientering med visse egenskaper og hun bruker en form for binærsøk: Til å begynne med er alle orienteringer mulige løsninger, men for hver iterasjon halveres antallet kandidater, helt til én gjenstår. Én iterasjon tar lineær tid som funksjon av antall kanter, m. Hva blir den totale kjøretiden? Oppgi svaret med Θ-notasjon. Forklar kort.

10% 17. Hvordan kan du effektivt finne asykliske og sterke orienteringer?

10% 18. Hvordan kan du effektivt finne k-orienteringer ved hjelp av maks-flyt?