Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4120 Algoritmer og datastrukturer

Faglig kontakt under eksamen	Magnus Lie Hetland
Telefon	918 51 949

Eksamensdato	10. desember, 2018
Eksamenstid (fra–til)	09:00–13:00
Hjelpemiddelkode/tillatte hjelpemidler	D
Annen informasjon	Oppgavearkene leveres inn, med svar i svarrute under hver oppgave

Målform/språk	Bokmål
Antall sider (uten forside)	7
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

Spørsmål til fagstaben?

Vi kan kun svare på forespørsler om mulige feil, mangler eller uklarheter i oppgaveteksten.

Vi går én runde og har begrenset tid. Les igjennom alle oppgavene først og ha evt. spørsmål klare!

Les dette nøye

(i) Les hele eksamenssettet nøye før du begynner!

(ii) Skriv svarene dine i svarrutene og levér inn oppgavearket. Bruk gjerne blyant! Evt. kladd på eget ark først for å unngå overstrykninger, og for å få en egen kopi.

(iii) Ekstra ark kan legges ved om nødvendig, men det er meningen at svarene skal få plass i rutene på oppgavearkene. Lange svar teller ikke positivt.

Oppgaver

5% 1. I en sammenhengende graf med $n \geq 1$ noder og m kanter, hva er det minste og største antall kanter et spenntre kan ha?

5% 2. I en Huffman-kode for et alfabet med $n \geq 1$ tegn, hva er den minste og største lengden et kodeord for ett tegn kan ha?

5% 3. Anta at Π er en forgjengermatrise for alle-til-alle-varianten av korteste vei-problemet. Hva representerer π_{ij}, om du antar $\pi_{ij} \neq \textsc{nil}$?

5% 4. Hva er et *snitt* (*cut*) i et flytnett (*flow network*) $G = (V, E)$?

5% 5. Hva er et nodedekke (*vertex cover*) for en graf $G = (V, E)$?

5% 6. TABLE-INSERT har i verste tilfelle kjøretid $\Theta(n)$, men *amortisert* kjøretid $O(1)$. Hva betyr det?

5% 7. I n-tårns-problemet har man et $n \times n$-sjakkbrett og skal plassere n tårn på brettet, så man har nøyaktig én brikke i hver rad og én i hver kolonne. (Evt. se grundigere forklaring på side 7.) Din venn Klokland har funnet en løsning på dette problemet, for én bestemt n – altså en bestemt plassering av de n brikkene. Din venn Smartnes vil vite hvilken av de mulige løsningene Klokland har funnet, men det vil han ikke fortelle henne. I stedet får hun prøve å gjette, ved å stille ja/nei-spørsmål. Hvor mange spørsmål trenger hun, i verste tilfelle, hvis du *ikke* kan anta noe om hvor smart Smartnes er? Oppgi svaret i asymptotisk notasjon. Forklar svaret.

5% 8. I 0-1-ryggsekkproblemet, anta at du har n gjenstander og en ryggsekk-kapasitet på m. Hva blir kjøretiden til pensumløsningen?

5% 9. Om du setter tallene $1, \ldots, n^3$ inn i et binært søketre i tilfeldig rekkefølge, der $n = 2^k - 1$ for et heltall $k \geq 1$, hva er den forventede høyden til treet som funksjon av k? Oppgi svaret i Θ-notasjon.

	1	2	3	4	5	6
Q	2	3	5	7	8	9

Figur 1: Eksempel-kø til oppgave 11. Her er verdiene fylt inn fra starten, så Q.*head* = Q.*tail* = 1

	1	2	3	4	5	6
Q	7	8	9	2	3	5

Figur 2: Eksempel-kø til oppgave 11. Her er verdiene *ikke* fylt inn fra starten. Q.*head* = Q.*tail* = 4

5% 10. Din venn Smartnes utfører DFS på en DAG. Hvilke typer kanter kan hun få?

(Her refererer *typer* til DFS sin kantklassifisering.)

5% 11. Din venn Gløgsund har satt inn n ulike tall i stigende rekkefølge i en FIFO-kø Q, implementert som en tabell. Køen er full, så tabellen inneholder kun elementene hun har satt inn. Elementene utgjør altså enten ett stigende segment, $\langle x_1, \ldots, x_n \rangle$ (se figur 1) eller to stigende segmenter, $\langle x_k, \ldots, x_n, x_1, \ldots x_{k-1} \rangle$ (se figur 2), der $x_i < x_{i+1}$ for $i = 1 \ldots n-1$. Dessverre har Gløgsund glemt hvor køen starter og slutter (dvs., Q.*head* og Q.*tail*). Beskriv en algoritme som lar henne finne ut dette, så effektivt som mulig. Hva blir kjøretiden?

(Her holder det at du gir en svært overordnet beskrivelse, uten å ta hensyn til praktiske detaljer eller spesialtilfeller.)

5% 12. I hver iterasjon av den ytterste løkka til INSERTION-SORT skal man utføre en operasjon på A$[1\,..\,j-1]$ og A$[j]$. Hva er denne operasjonen, og hva er kjøretiden som funksjon av j? Bruk O-notasjon.

W	1	2	3	4
1	0	1	5	6
2	∞	0	−3	4
3	∞	∞	0	2
4	∞	∞	∞	0

Figur 3: Vektmatrisen til G, brukt i oppgave 14

$T^{(1)}$	1	2	3	4
1	1	0	1	0
2	1	1	1	0
3	0	0	1	0
4	0	1	0	1

Figur 4: Forrige tilstand, brukt i oppgave 15

5% 13. Din venn Lurvik har utviklet en ny prioritetskø som han er veldig stolt av. Den kan konstrueres i lineær tid, akkurat som en binærhaug (*binary heap*), mens EXTRACT-MIN og DECREASE-KEY har kjøretid på henholdsvis $O(n^2)$ og $O(n^3)$, der n er antall elementer i køen. Hva blir kjøretiden til DIJKSTRA om man bruker Lurviks prioritetskø? Begrunn svaret kort.

(Merk at Lurvik her *ikke* bruker en serie med kall til INSERT for å bygge køen sin, men bygger den i lineær tid i starten av DIJKSTRA.)

5% 14. La G være en vektet graf, definert av vektmatrisen i figur 3. Utfør DAG-SHORTEST-PATH på grafen, med 1 som startnode.

Fyll ut avstandene til hver node etter hver iterasjon i hver rad i tabellen nedenfor.

	1	2	3	4
0	0	∞	∞	∞
1				
2				
3				
4				

5% 15. Du har utført én iterasjon av TRANSITIVE-CLOSURE, og endt opp med tabellen $T^{(1)}$ som vist i figur 4. Utfør neste iterasjon, og fyll inn de resulterende verdiene i tabellen nedenfor.

$T^{(2)}$	1	2	3	4
1				
2				
3				
4				

5% 16. Hva er kjøretiden til algoritmen ALPHA, nedenfor, der $n \geq 1$ er et heltall? Oppgi svaret i Θ-notasjon, som funksjon av n.

```
ALPHA(n)
for i = 1 to n
    for j = i to n
        for k = 1 to n
            print "Lurvik rulz!"
```

5% 17. Hva er kjøretiden til algoritmen BETA, nedenfor, der $n \geq 1$ er et heltall? Oppgi svaret i Θ-notasjon, som funksjon av n.

```
BETA(n)
if n ≥ 2
    m = ⌊n/2⌋
    BETA(m)
    BETA(n − m)
    for i = 1 to n^2
        print "Smartnes rocks!"
```

5% 18. Hvilket problem løser algoritmen DELTA, nedenfor, der $n, m \geq 0$ er heltall?

```
GAMMA(n, m)
if m == 0
    return n
else return GAMMA(n, m − 1) + 1
```

```
DELTA(n, m)
if m == 0
    return 0
else return GAMMA(n, DELTA(n, m − 1))
```

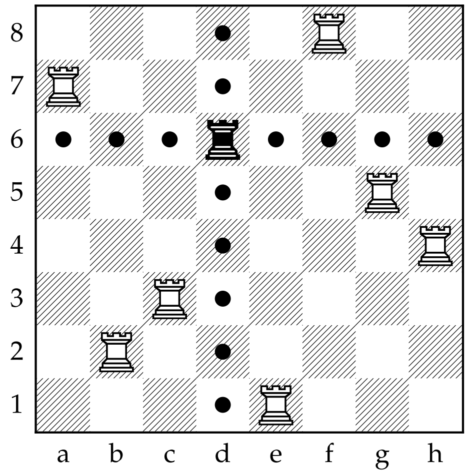
5% 19. Din venn Kvikstad skal gi navn til et sett med datamaskiner. Han vil at navnene skal begynne med ulike bokstaver, med ett navn på A, ett på B, osv., så langt han kommer. Han har ikke bestemt seg for hvilken maskin som skal få navn som begynner med hvilken bokstav, men han har noen navneforslag for hver maskin, og vil velge fra dem. Tegn opp et flytnett (på neste side) som finner et gyldig utvalg for ham, basert på følgende tabell:

Maskin	**Mulige navn**
1	Abhoth
2	Aletheia, Byatis
3	Aletheia, Byakhee, C'thalpa
4	Azathoth, Byakhee, Cthulhu, Dagon

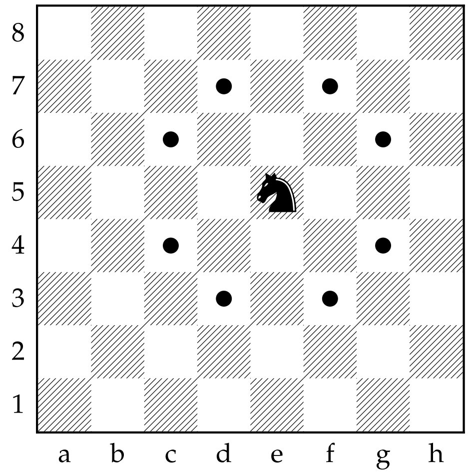
Svar på oppgave 19:

5% 20. En springer er plassert i rute a1 på et $n \times n$-sjakkbrett og skal flyttes k ganger, eller til et slikt trekk ville havnet utenfor brettet. Hver gang skal ett av de 8 mulige trekkene velges tilfeldig. Beskriv en algoritme basert på dynamisk programmering som beregner sannsynligheten for at springeren fortsatt er på brettet etter k trekk.

(Evt. se grundigere forklaring på neste side.)



Figur 5: n-tårns-problemet. Det svarte tårnet truer rutene angitt ved svarte prikker



Figur 6: En springer kan hoppe til en av åtte ruter, som angitt ved svarte prikker

Om sjakkproblemene

I problemene vi ser på tillater vi vilkårlig store kvadratiske rutenett som sjakkbrett, så et $n \times n$-sjakkbrett består av n rader og n kolonner.

I n-tårns-problemet (se oppgave 7) har vi et $n \times n$-sjakkbrett og n tårn som skal plasseres på dette brettet, så ingen av dem *truer* hverandre. Et tårn *truer* alle brikker i samme rad eller kolonne, uavhengig av farge. (Se figur 5 for et eksempel.) Med andre ord går problemet ut på å plassere nøyaktig én brikke i hver rad og nøyaktig én brikke i hver kolonne. Det er naturligvis flere korrekte løsninger på dette problemet.

I springerproblemet (se oppgave 20) har vi kun én brikke på et $n \times n$-sjakkbrett. Denne brikken er en *springer*, som kan flyttes til én av (maksimalt) åtte ulike posisjoner, som vist i figur 6. Det vil si, den kan flyttes to ruter opp, ned, til høyre eller venstre, og deretter én rute til siden. For eksempel to ruter til høyre og så én rute ned, eller to ruter ned og én til høyre.

Dersom springeren står nært kanten av brettet, vil noen av disse åtte trekkene ikke lenger være lovlige, siden de havner utenfor brettet. I problemet vårt velger vi likevel blant alle åtte (med uniform sannsynlighet), og vil se om springeren havner utenfor brettet. Hvis den for eksempel står på rute a1 (nederst til venstre) og flytter to trinn opp og ett til venstre, som er et av de åtte mulige trekkene, så har den havnet utenfor og vi kan ikke lenger flytte den videre.

Vi velger hvert trekk uavhengig av de andre. Husk at sannsynligheten for at to uavhengige hendelser A og B med respektive sannsynligheter $P(A)$ og $P(B)$ begge skal inntreffe er produktet av sannsynlighetene, $P(A) \cdot P(B)$.