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	9. august, 2017
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)	8
Antall sider vedlegg	0

Informasjon om trykking av eksamensoppgave

Originalen er

1-sidig ☑ **2-sidig** ☐

sort/hvit ☑ **i farger** ☐

Skal ha flervalgskjema ☐

Kvalitetssikret av Pål Sætrom

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 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.

(iv) 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.

(v) Eksamen har 15 oppgaver, totalt verdt 100 poeng. Poengverdi er angitt ved hver oppgave.

Oppgaver

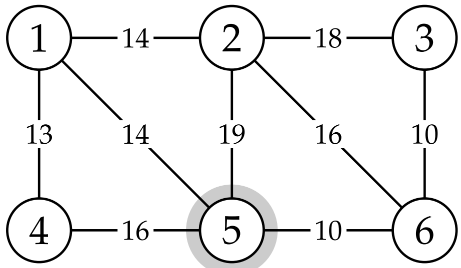
(5 p) 1. Hva er *worst-case*-kjøretiden til INSERTION-SORT? Oppgi svaret i Θ-notasjon.

(5 p) 2. Din venn Lurvik har et program som sorterer lenkede lister, med en tilpasset versjon av INSERTION-SORT. Hun har nettopp lært seg MERGE-SORT, og har lyst til å bruke den, men er bekymret for at den ikke vil være så effektiv om den tilpasses lenkede lister. Hva mener du? Forklar.

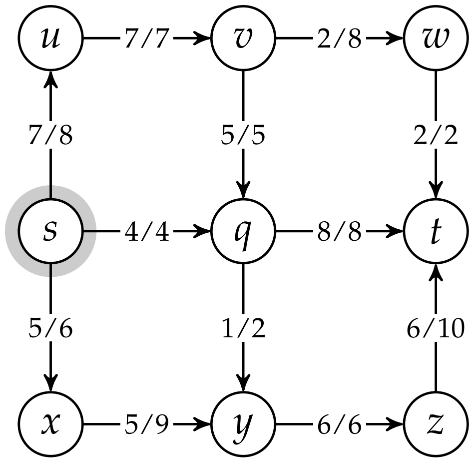
(5 p) 3. Du har oppgitt en tabell $A = \langle 1, 7, 2, 3, 5, 4, 6 \rangle$.

Utfør BUILD-MAX-HEAP(A) og deretter HEAP-EXTRACTMAX(A). Oppgi A etterpå.

Merk: I svaret så er $A.\mathit{heap\text{-}size} = A.\mathit{length} - 1$. Det spørres her om hele A, ikke bare heapen.



Figur 1: En vektet, urettet graf, brukt i oppgave 6



Figur 2: Flytnettverk brukt i oppgave 9

(5 p) 4. Hvilke to egenskaper ved et problem ser vi etter for å avgjøre om vi vil bruke dynamisk programmering (DP)? Oppgi først egenskapen som er nødvendig for at DP skal være *mer effektivt* enn naturlige alternativer, og deretter egenskapen som er nødvendig for at DP skal gi *korrekt svar*.

(5 p) 5. Du skal konstruere en Huffman-kode for tegnene a–d, med frekvenser som angitt nedenfor.

Tegn	a	b	c	d
Frekvens	4	1	6	2

Hva blir kodeordet for b?

Merk: Venstre barn plukkes ut først, og kanten til venstre barn får verdien 0.

(5 p) 6. Hvis du utfører MST-Prim på grafen i figur 1, med 5 som rot, hvilken kant vil velges som den femte i rekken? Det vil si, hvilken kant vil være den femte som legges til i løsningen?

Oppgi kanten på formen (i, j), der $i < j$.

(5 p) 7. Din venn Smartnes vil finne korteste vei fra node 1 til node 4 i grafen i figur 3 på neste side, og har bestemt seg for å bruke Dijkstra, med 1 som startnode. Oppgi avstandene han finner til nodene 2, 3 og 4, i rekkefølge.

(5 p) 8. Lurvik og Smartnes har slått seg sammen, for å finne de korteste veiene mellom *alle* noder i en rettet, vektet graf. De har valgt å bruke Floyd-Warshall, men etter å ha utført den manuelt i et par iterasjoner har de begynt å lure på om det er noe galt med grafen deres. Du har sagt du skal hjelpe dem med neste iterasjon.

$D^{(3)}$ og $\Pi^{(3)}$ er som angitt i figur 4 på neste side. Hva blir $D^{(4)}$ og $\Pi^{(4)}$?

Fyll ut tabellene nedenfor.

Merk: Vi antar her en implementasjon som i læreboka, det vil si at vi i hver iterasjon k lager nye tabeller $D^{(k)}$ og $\Pi^{(k)}$, heller enn en mer plass-effektiv variant som overskriver tabellene.

$D^{(3)}$

	1	2	3	4
1				
2				
3				
4				

$\Pi^{(3)}$

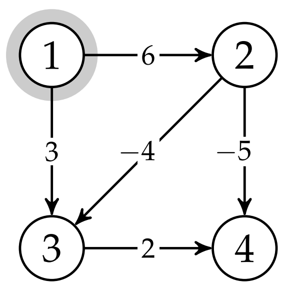
	1	2	3	4
1				
2				
3				
4				

(5 p) 9. Figur 2 på side 2 viser flytnettverket G, med kilde s, sluk t og flyt f. Er flyten maksimal? Svar ja eller nei.

Hvis ja, oppgi også mengden av noder som kan nås (dvs., som det finnes stier til) fra s i G_f.

Hvis nei, oppgi også nodene i en flytforøkende sti (*augmenting path*), i rekkefølge.

(5 p) 10. Stemmer det at $\mathrm{P} \subseteq \text{co-NP}$? Forklar svært kort.



Figur 3: En vektet, rettet graf til bruk i oppgave 7

$D^{(3)}$

	1	2	3	4
1	0	−3	5	−1
2	∞	0	∞	2
3	∞	−8	0	−6
4	∞	−5	3	−3

$\Pi^{(3)}$

	1	2	3	4
1	NIL	3	1	2
2	NIL	NIL	NIL	2
3	NIL	3	NIL	2
4	NIL	3	4	2

Figur 4: Forrige tilstand i utførelsen av FLOYD-WARSHALL, brukt i oppgave 8

(10 p) 11. Hvilke sorteringsalgoritmer i pensum har lineær forventet (*average-case*) kjøretid, under normale antagelser? Hva er disse antagelsene? (Det holder med et par stikkord per algoritme.)

(10 p) 12. En *uavhengig mengde* (*independent set*) i en graf $G = (V, E)$ er en delmengde $U \subseteq V$ av nodene som er slik at hver kant i E er tilkoblet maksimalt én node i U (dvs., ingen av nodene i U er naboer). *Størrelsen* til en uavhengig mengde er antall noder den inneholder. Vis at følgende problem er NP-komplett.

$$\text{INDEPENDENT-SET} = \{\langle G, k\rangle : G \text{ er en graf med en uavhengig mengde med størrelse } k\}$$

Det holder med en kort forklaring, men alle elementene i et NP-kompletthetsbevis må dekkes.

(10 p) 13. Du har oppgitt et sett med regler av følgende type, som beskriver en ukjent mengde S, der S er en delmengde av $\{1, \ldots, n\}$, for en gitt n:

«Hvis x_1 eller x_2 eller $\cdots$ eller x_{m-1} ligger i S så ligger x_m i S.»

Her er $x_1, \ldots, x_m$ elementer i $\{1, \ldots, n\}$, og m kan variere fra regel til regel. Anta at du får tid til å bygge en datastruktur basert på reglene. Deretter skal du effektivt kunne løse følgende problem:

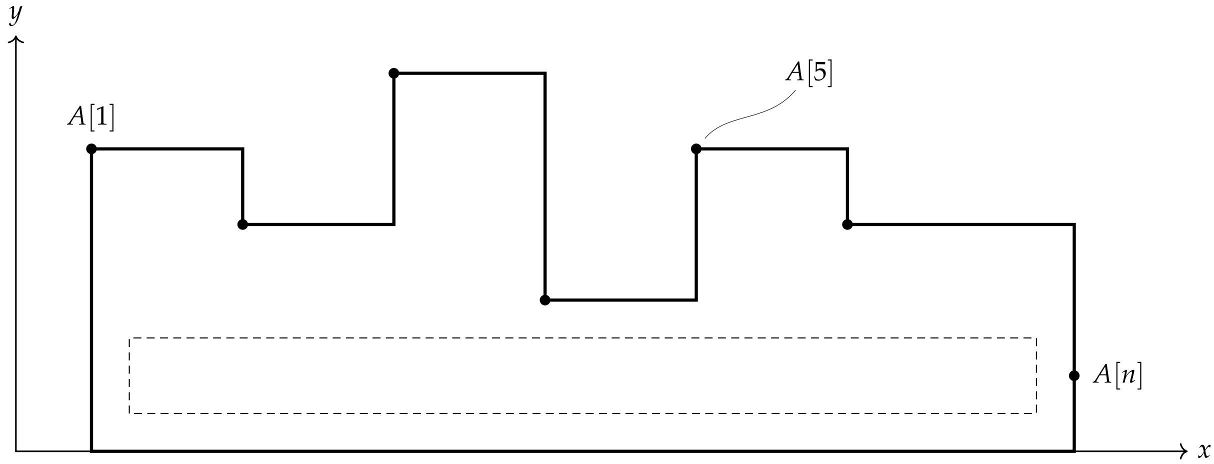
Input: En verdi x som skal ligge i S.
Output: Hele mengden S.

Her skal S være den *minste mengden* som tilfredsstiller opplysningene du har fått. Det vil si, S inneholder x og akkurat de elementene som kreves av reglene, men ingen andre. Du finner et eksempel på side 8.

Beskriv hvordan du vil løse dette problemet så effektivt som mulig.

(10 p) 14. Du har oppgitt en tabell $A[1 \,..\, n]$, der hvert element $A[i]$ er et punkt med positive koordinater (x_i, y_i) i planet, med $x_i < x_{i+1}$ for $i = 1 \ldots n-1$. Du skal finne et rektangel med horisontale/vertikale sider som ligger innenfor regionen definert av punktene, som vist i figur 5 på neste side. Mer spesifikt så skal du finne det *største* rektanglet av denne typen, altså det med størst areal.

Beskriv en algoritme som løser problemet så effektivt som mulig. Det holder med en overordnet, stikkordspreget forklaring, uten grundige implementasjonsdetaljer. For enkelhelts skyld trenger du bare finne *arealet* til løsningen, ikke koordinatene. Hva blir kjøretiden? (Svarrute på neste side.)



Figur 5: Eksempel på punkter, brukt i oppgave 14. Den tykke streken angir regionen, og består av en horisontal strek etterfulgt av en vertikal strek fra hvert punkt $A[i]$ til $A[i+1]$, for $i = 1 \ldots n-1$; fra $A[1]$ og $A[n]$ går det linjer ned til x-aksen, som utgjør den nederste siden av regionen. Det stiplede rektanglet er et eksempel på en gyldig, men ikke optimal løsning.

Svarrute til oppgave 14:

(10 p) 15. Du har oppgitt en todimensjonal tabell $A[1\,..\,n, 1\,..\,n]$ med reelle tall. Du skal velge ut ett tall fra hver rad slik at (1) ingen av de utvalgte tallene er i samme kolonne som det utvalgte tallet i neste rad, og (2) summen av tallene er størst mulig. (For $A[i, j]$ er i raden og j kolonnen.) Beskriv en algoritme som løser problemet så effektivt som mulig. Hva blir kjøretiden?

Eksempel til oppgave 13. Anta at du får oppgitt følgende regler:

(i) Hvis 1 eller 3 ligger i S så ligger 2 i S.

(ii) Hvis 2 ligger i S så ligger 3 i S.

(iii) Hvis 1 eller 3 ligger i S så ligger 4 i S.

Du får oppgitt et element x som ligger i S, og skal liste opp elementene i S. Her er svarene for ulike x:

$$
\begin{aligned}
x = 1 &\implies S = \{1, 2, 3, 4\} \\
x = 2 &\implies S = \{2, 3, 4\} \\
x = 3 &\implies S = \{2, 3, 4\} \\
x = 4 &\implies S = \{4\} \\
x = 5 &\implies S = \{5\}
\end{aligned}
$$

I hvert tilfelle inneholder altså mengden S det oppgitte elementet x, men også de elementene den *må* inneholde for at reglene (i)–(iii) skal gjelde. Merk at S ikke inneholder unødvendige elementer.