Avsluttende eksamen i TDT4120 Algoritmer og datastrukturer

Eksamensdato	14. desember 2011
Eksamenstid	1500–1900
Sensurdato	14. januar
Språk/målform	Bokmål
Kontakt under eksamen	Magnus Lie Hetland (tlf. 91851949)
Tillatte hjelpemidler	Ingen trykte/håndskrevne; bestemt, enkel kalkulator

! **Les alle oppgavene før du begynner, disponer tiden og forbered spørsmål til faglærer ankommer lokalet.** Gjør antagelser der det er nødvendig. Skriv kort og konsist på angitt sted. Lange forklaringer og utledninger som ikke direkte besvarer oppgaven tillegges liten eller ingen vekt.

Algoritmer kan beskrives med tekst, pseudokode eller programkode, etter eget ønske, så lenge det klart fremgår hvordan den beskrevne algoritmen fungerer. Korte, abstrakte forklaringer kan være vel så gode som utførlig pseudokode, så lenge de er presise nok. Kjøretider oppgis med asymptotisk notasjon, så presist som mulig.

Oppgave 1 (44%)

a) Anta at nodene i en rettet graf kan ordnes fra venstre mot høyre slik at alle kantene peker fra venstre mot høyre. Hva kalles en slik ordning?

Svar (4%):

b) Anta at den urettede grafen $G = (V, E)$ er sammenhengende. Hva kalles den minste delmengden F av E som er slik at (V, F) er sammenhengende? (Merk: Det kan være flere slike minste delmengder.)

Svar (4%):

c) Hvilket designprinsipp (dynamisk programmering, splitt og hersk, grådighet) brukes i Huffmans algoritme?

Svar (4%):

d) Hva er kjøretiden til QUICKSORT i verste tilfelle?

Svar (4%):

e) Hva slags kø brukes i DFS?

Svar (4%):

f) Hvilken algoritme vil du bruke for å finne korteste vei mellom to noder i en urettet, uvektet graf?

Svar (4%):

g) Å finne et minimalt snitt (*min-cut*) er ekvivalent med et annet pensumproblem. Hvilket?

Svar (5%):

h) Du vet at problem A er i NP og problem B er i NPC. Du vil vise at A også er i NPC. Hvilken vei vil du redusere?

Svar (5%):

i) Hvilken algoritme vil du bruke for å finne korteste vei mellom to noder i en rettet, vektet graf hvis du ikke kan anta noe om kantvektene?

Svar (5%):

Anta at du skal fordele spillere i et eller annet ballspill på to lag. Du har estimert hver spiller sine ferdigheter med et positivt reelt tall, og ønsker nå å lage så *ujevne* lag som mulig (det vil si at summen av ferdighetsverdiene i de to lagene skal være så forskjellige som mulig). Lagene skal ha like mange deltakere.

j) Hvordan kan du løse fordelingsproblemet med kjøretid $\Theta(n \lg n)$?

Svar (5%):

Oppgave 2 (35%)

a) Du skal lage en komplett, rettet graf med noder 1, 2, … , n. Hvis du kan velge retningen på hver av kantene fritt, hvor mange ulike grafer kan du konstruere?

Svar (7%):

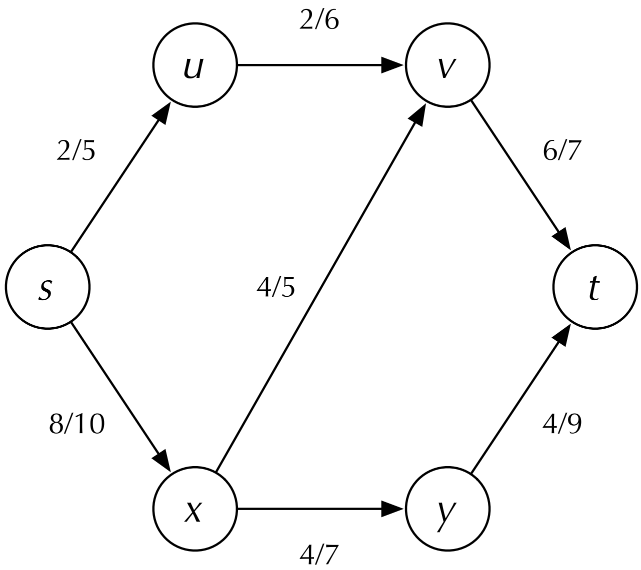
b) Du har to mengder **A** og **B** (begge med størrelse n) og et tall x. Beskriv en effektiv algoritme som lar deg avgjøre om det finnes et tall y i **A** og et annet tall z i **B** slik at $x = y + z$. Hva blir kjøretiden?

Svar (7%):

c) En venn av deg påstår han har utviklet en generell prioritetskø der operasjonene for å legge til et element, å finne maksimum og å ta ut maksimum alle har kjøretid $O(1)$ i verste tilfelle. Forklar hvorfor dette ikke kan stemme.

Svar (7%):

Betrakt følgende flytnettverk over nodene $\{s, t, u, v, x, y\}$, med kilde s og sluk t:



Flyt og kapasitet er angitt på kantene (for eksempel er flyten fra x til y på 4, med kapasitet på 7).

d) Angi den flytforøkende stien (*augmenting path*) som gir størst økning i flyten. Svaret oppgis som en sekvens av noder.

Svar (7%):

e) Gi (f.eks. tegn) et (lite) eksempel på at et korteste-vei-tre (for eksempel som produsert av Dijkstras algoritme) ikke er et minimalt spenntre.

Svar (7%):

Oppgave 3 (21%)

La T være en uvektet, urettet asyklisk graf (det vil si et tre uten spesifisert rot). La *diameteren* til T være lengden til (antall kanter i) den lengste stien i T.

a) Beskriv en effektiv algoritme for å beregne tre-diameter som beskrevet. Hva blir kjøretiden?

Svar (7%):

I et tre T med rot r så er en node v en *etterkommer* av u hvis (og bare hvis) det finnes en sti fra r til v som går innom u. Du ønsker nå for et gitt par med noder u og v å avgjøre om u er en etterkommer av v, om v er en etterkommer av u, eller om ingen av dem er etterkommere av hverandre. Vi kan kalle dette *etterkommerproblemet*.

b) Beskriv en algoritme som preprosesserer treet T med kjøretid $O(n)$ slik at etterkommerproblemet kan løses for T med kjøretid $O(1)$.

Svar (7%):

Du er teaterregissør, og skal lage en øvingsplan – en oversikt over hvilke skuespillere som skal møte på hvilke dager under innøvingen av et teaterstykke. Du har oppgitt følgende:

- Et sett med skuespillere, der skuespiller i er betalt for å møte på $D[i]$ øvingsdager (og altså ikke kan møte flere dager enn dette).
- Et sett med scener, der scene i skal jobbes med på $S[i]$ av øvingsdagene. Summen av $S[i]$ over alle scenene er lik antall dager tilgjengelig. (Kun én scene skal jobbes med per dag.)
- Informasjon om hvilke skuespillere som har anledning til å møte hvilke av dagene. (Enkelte er altså opptatt på noen av dagene.)
- Informasjon om hvilke skuespillere som er med i hvilke scener (og dermed må møte hvis scenen skal jobbes med).

Du ønsker å utarbeide en algoritme som lager en øvingsplan, det vil si, som avgjør hvilken scene det skal jobbes med på hver av øvingsdagene. Dersom det er umulig å få til en slik plan skal algoritmen oppgi dette.

c) Beskriv kort en algoritme som løser problemet. (Bruk gjerne en figur i svaret.)

Svar (7%):