Eksamen i

TDT4120 Algoritmer og datastrukturer

August 2005

Faglige kontakter under eksamen:
Magnus Lie Hetland, Arne Halaas
Tillatte hjelpemidler: Bestemt enkel kalkulator. Ingen trykte eller håndskrevne hjelpemidler.

Merk: Svarene skal skrives på angitt sted på oppgavearket.

Oppgave 1 (80%)

De følgende deloppgavene skal besvares med “ja” eller “nei” (sett kryss ved det som passer). Dersom deloppgaven er et utsagn (og ikke et spørsmål) betyr “ja” at utsagnet er sant og “nei” at utsagnet er usant eller ikke stemmer (evt. ikke gir mening).

En riktig besvart deloppgave gir 4 poeng, en ubesvart deloppgave gir 0 poeng og en galt besvart deloppgave gir –4 poeng. En negativ totalsum for hele denne oppgaven teller som 0 poeng når totalen for eksamen beregnes.

Les oppgavene nøye. Oppgavetekstene er utformet slik at svarene ikke skal være opplagte. Svar bare dersom du er sikker på at du har forstått oppgaven og at du vet svaret.

—

[ ] Ja [ ] Nei *a*) Uansett form på et binært tre med $n \geq 1$ interne noder vil antall løvnoder være $n + 1$.

—

[ ] Ja [ ] Nei *b*) Grådighet er en generalisering av dynamisk programmering.

—

[ ] Ja [ ] Nei *c*) Warshalls algoritme kan brukes når man har kanter med negativ vekt i grafen.

—

Du har oppgitt en graf $G = (V, E)$ og ønsker å finne det største settet $V' \subseteq V$ der alle nodene i V' har kanter til alle de andre andre nodene i V'.

[ ] Ja [ ] Nei *d*) Problemet kan effektivt løses med dynamisk programmering.

—

[ ] Ja [ ] Nei *e*) Det er umulig å lage en algoritme for å sjekke om (vilkårlige) andre algoritmer er korrekte.

—

Vi har gitt en graf $G = (V, E)$, der $V = \{1, \dots, 7\}$ og $E = \{(1, 2), (1, 4), (2, 4), (2, 5), (3, 1), (3, 6), (4, 3), (4, 5), (4, 6), (4, 7), (5, 7), (7, 6)\}$.

Vi har også gitt kantvektfunksjonen $w(u, v)$, gitt ved
$w(1, 2) = 2$, $w(1, 4) = 1$, $w(2, 4) = 3$, $w(2, 5) = 10$, $w(3, 1) = 4$, $w(3, 6) = 5$, $w(4, 3) = 2$, $w(4, 5) = 2$, $w(4, 6) = 8$, $w(4, 7) = 4$, $w(5, 7) = 6$, $w(7, 6) = 1$.

Simuler (utfør) Dijkstras algoritme på denne grafen, med 3 som start-node. Før første iterasjon er det kun node 3 som har riktig avstandsestimat. Hvis du kan velge mellom to noder, velg alltid den med lavest nummer.

[ ] Ja [ ] Nei *f*) Node 4 har riktig avstandsestimat etter andre iterasjon.

[ ] Ja [ ] Nei *g*) Node 5 har riktig avstandsestimat etter fjerde iterasjon.

[ ] Ja [ ] Nei *h*) Nodene i grafen G kan ordnes topologisk i $O(|E|)$ tid.

I den følgende deloppgaven, se bort fra kantretningene (det vil si, se på G som en urettet graf). La T være et minimalt spenntre for grafen G, beregnet med Prims algoritme med node 1 som startnode. Hvis du kan velge mellom to noder, velg vilkårlig.

[ ] Ja [ ] Nei *i*) Kanten (2, 4) er med i T.

—

Du har en hashfunksjon definert ved $h(k) = (7k + 3) \bmod 5$, der k er en heltallsnøkkel og mod er *modulus*- eller restoperatoren. For eksempel vil $h(1) = 0$. Du har også en hashtabell A med 5 plasser, $A[0], \dots, A[4]$. Du bruker lukket hashing (*linear probing*).

[ ] Ja [ ] Nei *j*) Etter å ha satt inn tallene 11, 13, 15 og 21 kan 14 settes inn uten kollisjon.

—

Betrakt rekurrensen $T(n) = 2T(n/2) + n$, $T(1) = 1$.

[ ] Ja [ ] Nei *k*) Rekurrensen er typisk for algoritmer basert på dynamisk programmering.

[ ] Ja [ ] Nei *l*) $T(\mathrm{n}) \in \Omega(n^2)$

[ ] Ja [ ] Nei *m*) $T(\mathrm{n}) \in O(n^2)$

—

I enkelte sanger (som “Gubben og gamla” og “The Twelve Days of Christmas”) øker lengden på versene med $\Theta(1)$ for hvert vers. Anta at antall vers er n.

[ ] Ja [ ] Nei *n*) Den totale sanglengden er $\Omega(n^{1.7})$.

—

En algoritme tar et vilkårlig heltall n som input og kjøretiden er $\Omega(n^{1.7})$.

[ ] Ja [ ] Nei *o*) Algoritmen har (nødvendigvis) eksponensiell kjøretid, som funksjon av problemstørrelsen.

—

[ ] Ja [ ] Nei *p*) Hvis en kant e er den billigste/letteste kanten i en graf så finnes det et minimalt spenntre som inneholder e.

—

[ ] Ja [ ] Nei *q*) Et minimalt spenntre er unikt hvis og bare hvis alle kantvektene er forskjellige.

—

[ ] Ja [ ] Nei *r*) Sortering ved sammenligning kan ikke løses med bedre kjøretid enn $\Omega(n \log n)$ i verste tilfelle.

—

[ ] Ja [ ] Nei *s*) Ingen av sorteringsalgoritmene i pensum har kjøretid $O(n)$ i verste tilfelle.

—

Du har n forskjellige verdigjenstander, hver med en vilkårlige positiv heltallsverdi (pris), og ønsker å finne ut om det er mulig å fordele gjenstandene likt på to av dine venner (så begge vennene mottar samme totale verdi).

[ ] Ja [ ] Nei *t*) Det finnes ingen kjent metode som kan avgjøre dette med en kjøretid bedre enn $O(n^2)$.

Oppgave 2 (10%)

Det er NP-komplett å finne den lengste stien i en graf. Anta at nodene i grafen $G = (E, V)$ er sanger og at kanten (u, v) finnes i E dersom sang v er en nyinnspilling ("cover") av sang u. Anta at nodene i grafen $G' = (E', V')$ er artister og at kanten (u', v') finnes i E' dersom artist v' har gjort en nyinnspilling av en av artist u' sine sanger. Anta at du kan ha vilkårlig mange sanger og artister.

a) (5%) Enten (1) argumenter for at det er NP-komplett å finne den lengste stien i G, eller (2) beskriv kort en effektiv algoritme som finner den lengste stien i G (angi kjøretid).

Svar:

b) (5%) Enten (1) argumenter for at det er NP-komplett å finne den lengste stien i *G'*, eller (2) beskriv kort en effektiv algoritme som finner den lengste stien i *G'* (angi kjøretid).

Oppgave 3 (10%)

I det følgende, anta at $G = (V, E)$ er en urettet, asyklisk graf der alle kantene i E har vekt 1. Hvis du finner de korteste avstandene fra en node v i V til alle de andre nodene i V så er *eksentrisiteten* til v den *lengste* av disse avstandene.

a) (4%) Beskriv kort en effektiv algoritme som finner en eller flere noder med minimal eksentrisitet i grafen G. Oppgi kjøretid i verste tilfelle (så presist som mulig) med asymptotisk notasjon. Begrunn svaret.

b) (3%) Hvor mange noder kan inngå i svaret til algoritmen i *a*)? Gi en kort begrunnelse.

c) (3%) Beskriv kort et praktisk eksempel der en slik algoritme kan være av interesse.