

Avsluttende eksamen i TDT4120 Algoritmer og datastrukturer

Eksamensdato	7. august 2009
Eksamenstid	0900–1300
Sensurdato	28. august
Språk/målform	Bokmål
Kontakt under eksamen	Magnus Lie Hetland (tlf. 91851949)
Tillatte hjelpemidler	Ingen trykte/håndskrevne; bestemt, enkel kalkulator

Vennligst les hele oppgavesettet før du begynner, disponer tiden og forbered evt. spørsmål til faglærer kommer til eksamenslokalet. 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 (30%)

a) Fyll inn følgende tabell. (Det holder med stikkord i problemkolonnen.)

Svar (20%):

Algoritme	Problem + evt. ekstrakrav til problem	Kjøretid
DIJKSTRA	<u>Korteste vei, én til alle, pos. vektor</u>	<u>$\Theta(E \lg V)$</u>
HUFFMAN	_____	_____
KRUSKAL	_____	_____
EDMONDS-KARP	_____	_____
FLOYD-WARSHALL	_____	_____
COUNTING-SORT	_____	_____

b) Løs følgende rekurrens. Oppgi svaret i asymptotisk notasjon. Begrunn svaret kort.

$$T(n) = 2T(n/3) + 1/n$$

Svar (10%):

Oppgave 2 (24%)

a) Hvilken grunnleggende algoritme brukes når man skal finne sterke komponenter?

Svar (8%):

b) Hva er fordelene med red-black-trees i forhold til vanlige binære søketrær?

Svar (8%):

c) Hva er fordelene med B-trær i forhold til red-black-trees?

Svar (8%):

Oppgave 3 (23%)

Vi kjører BFS fra node s i den uvektede grafen G . Det er en kant e mellom nodene u og v i G , men denne kanten er ikke med i bredde-først-treet. La d være avstandstabellen fra BFS.

a) Argumenter kort for at differansen mellom $d[u]$ og $d[v]$ ikke kan være mer enn 1.

Svar (10%):

Det kan være rimelig å tenke at negative sykler kan gi «uendelig korte» stier. Men: korteste-vei-algoritmer leter etter *simple paths*, altså stier *uten sykler*.

b) Hvorfor har vi likevel problemer med å finne korteste vei når en graf inneholder negative sykler?

Svar (6%):

Du har en sortert sekvens med n unike ID-nummer (positive heltall). Du vil finne det første (laveste) ledige ID-nummeret.

Eksempel: I sekvensen [1, 2, 3, 5, 7, 8] er 4 det første ledige nummeret.

c) Beskriv en effektiv algoritme for å finne det første ledige ID-nummeret. Hva blir kjøretiden? Begrunn svaret svært kort.

Svar (7%):

Oppgave 4 (23%)

Du har en streng med lengde n (og konstant alfabetstørrelse) lagret i en tabell med samme størrelse.

a) Beskriv en effektiv algoritme for å reversere strengen med konstant lagringsplass ut over strengen i seg selv (dvs *in-place*). Hva blir kjøretiden? Begrunn svaret kort.

Svar (10%):

Du har en liste L med n strenger (med konstant alfabetstørrelse) av varierende lengde (maks-lengde m ; datasettet inneholder totalt M tegn) og ønsker å avgjøre hvorvidt listen inneholder minst ett strengpar A, B , der A er et anagram for B (det vil si, A består av de samme tegnene som B , men i en annen rekkefølge).

(forts.)

- a) Beskriv en effektiv algoritme for å finne ut om L inneholder et slikt ordpar. Hva blir kjøretiden? Begrunn svaret kort.

Svar (7%):

Et *addagram* er en serie med ord av økende lengde, der hvert ord kan gjøres til et anagram av det forrige ved å fjerne en bokstav. For eksempel:

to, sto, stor, store, sorten, stormen, matrosen, samordnet,
motstander, demonstrant, motstanderne, demonstranten,
demonstrantene, demonstrantenes

Du ønsker nå å finne lengden på den lengste sekvensen av slike ord som finnes i ordlisten L . (Merk at ordene godt kan forekomme i en annen rekkefølge i ordlisten enn i sekvensen.)

- b) Beskriv en effektiv algoritme for å finne det lengste addagrammet i L . Hva blir kjøretiden? Begrunn svaret kort.

Svar (6%):