

EKSAMENSOPPGAVE

Fag: IAI20102 – Algoritmer og datastrukturer
Lærer: André A. Hauge

Dato: 3.6.2005	Tid: 0900-1300
Antall oppgavesider: 5 med forside	Antall vedleggssider: 0
Hjelpemidler: Alle trykte og skrevne hjelpemidler, kalkulator	
Sensurfrist: 24.6.2005	
KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG	

Råd for oppgaveløsningen:

- Les alle oppgavene før du starter.
- Dersom du finner at oppgaveteksten ikke gir tilstrekkelig informasjon, eller at oppgaveteksten er tvetydig, så må du gjøre dine egne forutsetninger. Disse må du i så fall presisere og begrunne.
- Dersom du får knapp tid, er det bedre å lage skisser på punktene du ikke får tid til fremfor å la punktene være blanke.
- Dersom du har mye tid igjen, ikke gå før tiden, disse 4 timene og hvordan du benytter deg av dem bestemmer din karakter i faget.
- Husk at begrunnelsen og beskrivelsen i svarene er svært viktig og derfor bør være klar.

Del 1 – Totalt 20 %

Oppgave 1.1 - 15 %

Under er det beskrevet noen operasjoner på ulike datastrukturer (en tabell er ekvivalent med en Java array). *For hver av dem skal du angi arbeidsmengden i O-notasjon, forutsatt at det ikke brukes ekstra hjelpestrukturer. Gjør også rede for eventuelle antagelser og presiseringer.*

1. Finne et bestemt element, for eksempel det 7. elementet, i en tabell.
2. Finne et bestemt element, for eksempel nr. 7, i en lenket liste.
3. Skrive ut alle elementene i en enkeltlenket liste i omvendt orden (det siste elementet skrives ut først etc.).
4. Finne et element med en gitt nøkkel i en usortert tabell.
5. Finne et element med en gitt nøkkel i en dobbeltlenket liste.
6. Finne et element med en gitt nøkkel i et binært søketre.
7. Sortere en enkeltlenket liste med reelle tall (float/double).
8. Sortere en tabell med 4242 heltall som alle ligger i intervallet $[0 - 42]$.
9. Sortere en tabell med heltall der du vet at 50 % av tallene er like.
10. Finne en record med en bestemt nøkkel i en hashtabell.

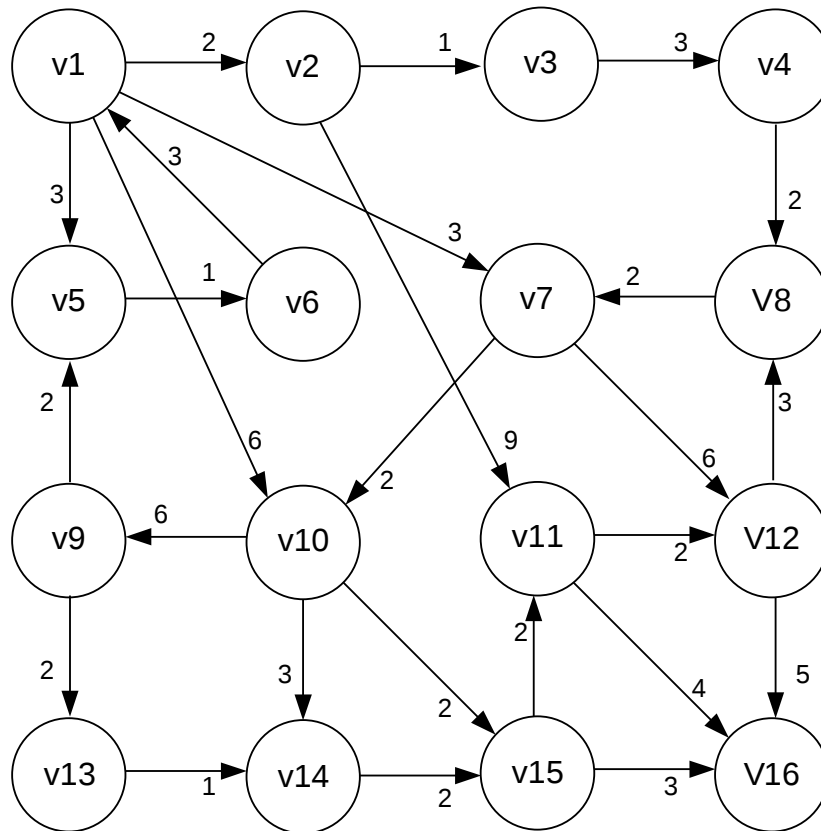
Oppgave 1.2 - 5 %

Hvis to algoritmer har samme O-notasjon vil det da si at de er like raske? Forklar eventuelt hvorfor/hvorfor ikke?

Slutt del 1

Del 2 – Totalt 20 %

I Figur 1 ser du en rettet vektet graf. Med grafer kan man løse mange forskjellige problemstillinger. Grafen nedenfor er ikke knyttet til noen bestemt case men representerer bare et tenkt nettverk av noder.



Figur 1 - En rettet vektet graf

Oppgave 2.1 - 15 %

I grafen ovenfor finn korteste vei fra node v1 til v16 med hensyn på kostnad. Benytt deg av den algoritmen som du synes passer best, tegn og forklar hva som skjer for hvert trinn i algoritmen.

Oppgave 2.2 - 5 %

Dersom du antar at Figur 1 er et representativt utsnitt av en stor graf og at hele grafen består av 2000 noder fra v1..v2000, hvordan ville du representert disse dataene i en implementasjon. Begrunn valg.

Slutt del 2

Del 3 – Totalt 30 %

I denne delen er det litt blandede oppgaver og du vil få bruk for alt det du har lært i kurset i den forstand at du skal velge den riktige algoritmen eller datastrukturen for å løse et bestemt problem. Du må også vise at du kjenner til de forskjellige datastrukturene sine sterke og svake sider samt vise at du behersker enkelte datastrukturer.

Oppgave 3.1 – 10 %

Hva er fordeler og ulemper med følgende datastrukturer når det gjelder søking, innsetting og fjerning av data:

- a) LinkedList
- b) HashMap
- c) BinarySearchTree
- d) Array

Oppgave 3.2 - 10 %

Hvilken datastruktur ville du brukt i følgende situasjoner:

- a) Du skal utvide funksjonaliteten til en teksteditor slik at den gjør stavekontroll etter hvert som brukeren skriver ord.
- b) Du skal programmere en metode, du har en rekursiv løsning som er svært effektiv og elegant, problemet er at programmeringsspråket du må benytte ikke støtter rekursjon. Hvordan kan du simulere rekursjon og likevel bruke algoritmen som du har med få omskrivninger?

Oppgave 3.3 - 5 %

Anta at du har en hashtabell med tabellsørrelse 11. Dersom du bruker open-addressing med linear probing ved kollisjon, hvordan vil følgende data elementer bli satt inn i tabellen:

6 5 28 17 18 115 16 7 71 60

Bruk hashfunksjonen $h(k) = k \bmod 10$

Oppgave 3.4 - 5 %

I et komplett binært tre med 20 noder der roten er definert til å være på nivå 0, hvor mange noder er det på nivå 4.

Slutt del 3

Del 4 – Totalt 30 %

Et firma som heter EternityPhoneSupport har ansatt deg for å løse en del problematikk angående telefon support systemet sitt.

Å ringe support i en sammenheng kan ta tid. Det er et begrenset antall operatører som skal håndtere en mengde henvendelser fra kunder. Vi antar at kundene har forskjellig prioritet, altså en V.I.P. kunde skal slippe å vente like lenge som andre i køen. Videre skal en operatør ha all tilgjengelig kundeinformasjon dersom det finnes i systemet når han tar imot en samtale.

Oppgave 4.1 - 30 %

I denne oppgaven skal du ved hjelp av figurer og tekst skissere et program som håndterer dette. Systemet skal gjøre følgende:

- Håndtere operatører: de er enten ledig | opptatt | pause
- Håndtere kunder. Du kan anta at når en kunde ringer supporttelefonen kan han gi informasjon om kundenummer via tastene på telefonen. Kundenummeret brukes til å finne nødvendig lagrede informasjon slik som hvem det er og eventuelt V.I.P. status. Dersom ikke kundenummer blir oppgitt behandles kunden generelt.
- Da en operatør er ledig og det finnes kunder i køen skal disse kobles sammen.

Det som er viktig i denne oppgaven er å skissere en løsning på hvordan problemet med et supporttelefonsystem kan løses, husk å få frem hvilke datastrukturer og algoritmer som er best for denne problemstillingen. Anta at det er mange operatører og mange kunder. Du kan anta at telefonsvarer systemet som kunden møter idet han/hun ringer support er laget og kjenner til systemet du lager og kaller den/de rette metodene for å registrere telefonhenvendelsen.

Begrunn alle valg, eventuelle antagelser, begrensninger eller forenklinger du gjør.

Slutt del 4

Lykke til og god sommer!

André