



Høgskolen i Gjøvik

Avdeling for elektro- og allmennfag

EKSAMEN

FAGNAVN: Videregående algoritmiske metoder

FAGNUMMER: V 135 A

EKSAMENS DATO: Mandag 3. mars 1997

KLASSE: 94HINDA / 94HINDE

TID: 09.00-13.00

FAGLÆRER: Frode Haug

ANT. SIDER UTLEVERT: 3 (inkludert denne forside)

TILLATTE HJELPEMIDLER: Alle trykte og skrevne.

- Kontroller at alle oppgavearkene er tilstede.
- **INNFØRING MED PENN**, evt. trykkblyant som gir gjennomslag.
Pass på at du ikke skriver på mer enn ett innføringsark om gangen
(da det blir uleselige gjennomslag om flere ark ligger oppå hverandre når du skriver).
- Ved innlevering skilles hvit og gul besvarelse og legges i hvert sitt omslag.
Oppgavetekst, kladd og blå kopi beholder kandidaten.
- Ikke skriv noe av din besvarelse på oppgavearkene.
- Husk kandidatnummer på alle ark (ikke oppgavearkene).

Oppgave 1 (teori, 35 %)

Denne oppgaven inneholder fire uavhengige oppgaver fra kap. 22, 23 og 30 i læreboka og fra artikkelen om «Burrows-Wheeler Transform»:

- a)** Vis konstruksjonsprosessen når bokas metode for Huffman-koding (s.324-330) brukes på teksten:
«FRODE FRODE FRODE KOMMODE FIRE BEIN OG MYE RART I HODET»
(inkludert de blanke). Hvor mange bits trengs for å kode denne teksten ?
Dvs. skriv/tegn opp:
- tabellen for bokstavfrekvensen (jfr. fig.22.3).
 - tabellen for «dad»en (jfr. fig.22.6).
 - Huffmans kodingstreet/-trien (jfr. fig.22.5 og koden øverst s.328).
 - bokstavenes bitmønster, med «code» og «len» (jfr. fig.22.7 og koden øverst s.329).
 - totalt antall bits som brukes for å kode teksten.
- b)** For å få det krypterte budskapet «SFTCPVWPBZFWPANLF» er metoden for «Vigenere cipher» (s.336) anvendt. Det er brukt mønsteret «CAB» (repetert så mange ganger som nødvendig) som nøkkel. Alfabetet som det krypteres etter, består av bokstavene A-Z, med ' ' (blank) som det første tegnet. Finn original-budskapet.
- c)** Følgende kanter i en (ikke-retted, ikke-vekted) graf er gitt:
AB CD GH BE DG HF BC DE AF CE AI EF EG FJ HI
- Vi skal nå utføre union-find på denne grafen. Tegn opp arrayen «dad»s innhold etterhvert som skogen bygges opp (jfr. fig.30.6) vha. «find»-funksjonen s.444 i læreboka. Ut fra dette: tegn også opp den resulterende union-find skogen (jfr. nedre høyre skog i fig.30.5).
- d)** Denne oppgaven omhandler Burrows-Wheeler transformasjon. Etter å ha satt opp venstre-rotasjonsmatrisen for originalteksten F, og deretter sortert den, så har L blitt: «KSURIT», med `primary_index = 5`. Finn T og deretter originalteksten F.

Oppgave 2 (koding, 35 %)

Vi skal nå se på noen typer rettede grafer, der alle typene er slik at alle noder har maksimalt en etterfølger, men gjerne flere forgjengere. Dvs. fra hver node går det maksimalt en kant ut, men gjerne flere kanter inn. Blant disse grafene finnes blant annet vanlige lister, enkle sykliske strukturer (løkker) og rotrettede trær (der alle nodene har peker opp til «mora», og ikke ned til barna). Det finnes imidlertid også mange andre grafer innenfor denne rammen.

Vi skal anta at vi har en slik graf representert ved noder av typen:

```
struct node {
    node* etterfølger;      // = NULL om ingen etterfølger finnes.
    int merke;
    // Diverse andre data i noden.
};
```

Det er N noder, og for å få tilgang til nodene har vi en array «node* graf[N+1]» (der vi bruker indeksene 1-N) som peker til alle de N nodene i tilfeldig rekkefølge.

Du skal lage to funksjoner som returnerer ‘false’ eller ‘true’ alt ettersom om:

a) grafen er en eneste stor sirkel (sykel/løkke).

b) grafen er ett rotrettet tre.

Beskriv også tankegangen du legger til grunn **for hver av dine to algoritmer**.

Det er altså en int-variabel «merke» i hver node, og denne antar vi er ‘0’ når funksjonene starter, og funksjonene kan bruke den som de vil. Forsøk å klare deg med å bruke så få forskjellige verdier som mulig for denne variabelen, eller helst å ikke bruke den i det hele tatt.

Oppgave 3 (koding, 30%)

Vi har et sjakkbrett som består av NxN ruter, der N kan ha verdiene 1, 2, 8. Lag et program (med en rekursiv prosedyre) som forsøker å plassere et minimum antall dronninger på dette NxN brettet, der dronningene tilsammen dekker alle rutene på brettet. Skriv ut en (eller flere) av disse minimumsløsningene.

Angående at en dronning «dekker» ruter: Når en dronning er plassert i en rute, så «dekker» den alle rutene på den linje/rad og kolonne den er plassert. I tillegg dekker den alle diagonalene (opptil 4 stk) ut fra vedkommende rute.

Lykke til !!

FrodeH