

# Hovedtrækkene i eksamenssættet i DM02

Kim Skak Larsen

Onsdag den 2. januar 2002, kl. 9–13

Dette er ikke en såkaldt standardbesvarelse. I en egentlig besvarelse af sættet skal der i nogle spørgsmål skrives betydeligt mere og grundigere end her. Nedenfor gives blot de essentielle elementer, der indgår. Der tages forbehold for trykfejl.

## Opgave 1 (25%)

**Spørgsmål a:**  $m = \frac{3}{2}n$ . □

**Spørgsmål b:** Der er nogle detaljer omkring, hvordan spidserne behandles, men det vigtige er, at man arbejder med et cirkeludsnit for hvilket, alle de korteste afstande er fundet. Dette udsnit udvides så gradvist til den komplette løsning. Det er vigtigt for korrektheden, at man i hvert skridt udvider sit udsnit i den retning, hvor afstanden er kortest fra den givne knude (kilden). □

**Spørgsmål c:** Det, man skal bruge, er, at en kant med størst vægt i en kreds kan “slettes”, og så vil der stadig være en korrekt løsning blandt de resterende kanter. En lille spids udgør en kreds med tre kanter. Løb alle spidserne igennem og “slet” en kant i hver. Nu er der præcis én kreds tilbage. Denne går hele vejen rundt enten den ene vej eller den anden vej om spidserne. Desuden kan der være små sidegrene på kredsen. “Slet” en tungeste kant i denne kreds. De tilbageværende kanter udgør nu et letteste udspændende træ. □

## Opgave 2 (25%)

**Spørgsmål a:** ’1’-tallene inden i et  $mc$  kald svarer til, at vi fjerner det første tegn i den tilsvarende streng. ’1’-tallene, der adderes til et kald af  $mc$ , optræder pga. at et ’?’ må indsættes for at matche første tegn i en af strengene. □

**Spørgsmål b:**  $mc("ab", "cd", 0, 0)$  giver anledning til kaldene  $mc("ab", "cd", 0, 1)$  og  $mc("ab", "cd", 1, 0)$ .

$mc("ab", "cd", 0, 1)$  giver anledning til kaldene  $mc("ab", "cd", 0, 2)$  og  $mc("ab", "cd", 1, 1)$ .

$mc("ab", "cd", 1, 0)$  giver anledning til kaldene  $mc("ab", "cd", 1, 1)$  og  $mc("ab", "cd", 2, 0)$ .

Man ser, at der er to kald af  $mc("ab", "cd", 1, 1)$ . □

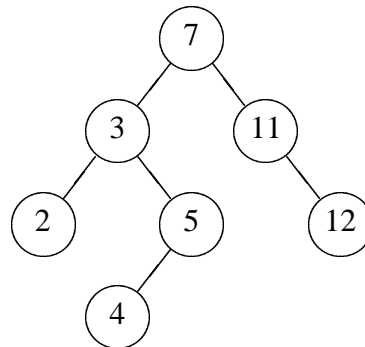
**Spørgsmål c:** Brug en af standardmetoderne: lav tabel, der dækker alle mulige måder, metoden kan kaldes på, medtag denne i metode-hovedet samt alle kald. Check først i metoden, om tabellen allerede indeholder svaret. Hvis ikke, beregnes det præcis som i den oprindelige kode bortset fra, at resultatet gemmes i tabellen i stedet for, at man returnerer. Til sidst i metoden returnes resultatet ved opslag under den korrekte tabelindgang. □

**Spørgsmål d:** Tabelstørrelsen er  $\Theta(|x| \cdot |y|)$ .

Kompleksiteten er den samme, da der kun bruges konstant-tids operationer til at fylde en given tabelindgang ud bortset fra rekursive kald, men disse fylder andre tabelindgange ud og skal derfor ikke tælles med under den givne tabelindgang. □

### Opgave 3 (25%)

**Spørgsmål a:**



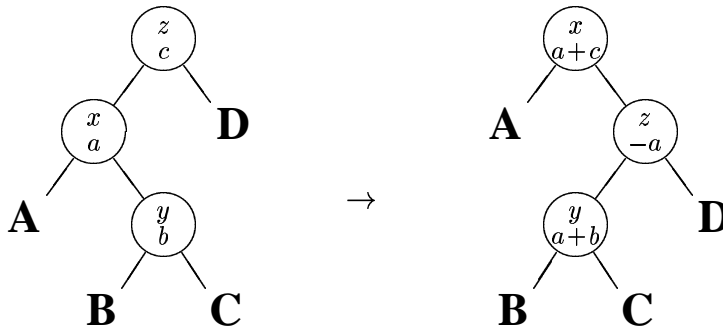
□

**Spørgsmål b:** *search*: Man medtager blot på vejen ned i træet et tal, der er summen af alle forskydningstallene på stien ovenover. Dermed kan man altid regne ud, hvad nøglen er, og det er alt, hvad man skal bruge, når man søger i et søgetræ.

*insert*: Søgning som ovenfor. Når man finder indsættelsesstedet, laves en ny knude. Den kan man f.eks. give nøglen som basistal og den negerede sum af forskydnindstallene fra stien ovenover som forskydningstal.

*add*: Læg  $c$  til rodens forskydningstal. □

**Spørgsmål c:**



□

**Opgave 4 (25%)**

**Spørgsmål a:** H er 18, 17, 15, 12, 7, 3. □

**Spørgsmål b:** Induktion i antallet af gange, man kommer til starten af while-løkken.

Første gang er  $i = n - 2$ , så mængden i invarianten er  $\{n - 1\}$ . Det skal altså vises, at  $H[n - 1] = A[n - 1]$ , men det gælder per initialisering. Da  $A$  er ikke-tomt, er  $n \geq 1$ , så  $i = n - 2 \geq -1$ .

I induktionsskridtet ses, at kun  $H[i' + 1]$  ændres, hvor  $i' = i - 1$  er den nye værdi af  $i$  efter hele while-kroppen er udført denne gang.

Dvs. at for  $k \geq i + 1 = i' + 2$  har  $H[k]$  per induktion den korrekte værdi. Mht.  $H[i' + 1]$  har vi fra while-kroppen, at  $H[i' + 1] = H[i] = H[i + 1] + A[i]$ . Per induktion er  $H[i + 1] = \sum_{j=i+1}^{n-1} A[j]$ , så  $H[i' + 1] = (\sum_{j=i+1}^{n-1} A[j]) + A[i] = \sum_{j=i}^{n-1} A[j]$ .

Da løkke-betingelsen er  $i \geq 0$ , er  $i' = i - 1 \geq -1$ . □

**Spørgsmål c:** Udtrykket  $i$  kan bruges som termineringsfunktion med grænsen  $-1$ . Funktionen starter over den værdi, kan ikke komme under (invarianten) og aftager med én hver gang. □

**Spørgsmål d:** Programmet terminerer, så det er korrekt, hvis postbetingelsen er opfyldt, når programmet terminerer. Efter while-løkken gælder invarianten, og udtrykket  $i \geq 0$  er falsk. Altså er  $-1 \leq i < 0$ , så da  $i$  er heltallig, er  $i = -1$ .

Ved indsættelse af  $i = -1$  i invarianten fås postbetingelsen, bortset fra det tilføjede  $\wedge (i \geq -1)$ . Men  $p \wedge q \Rightarrow p$  for alle logiske udsagn  $p$  og  $q$ , så det derivede udtryk medfører postbetingelsen.  $\square$