

DM507 – Opgaver uge 9

Recap fra forelæsninger: der er mindst tre naturlige niveauer at beskrive en algoritme på:

1. Som idé beskrevet i ord og tegninger.
2. Som pseudo-kode.
3. Som færdig kode (i f.eks. Java).

Dette er også de naturlige stadier at passere, når man skal udvikle algoritmer til et program i den virkelige verden. Hovedideen i en algoritme er allerede på plads på niveau 1, og korrekthedsanalyse og asymptotisk analyse med grovsortering af algoritmer efter voksehastighed kan normalt finde sted der. Niveau 2 tilføjer flere detaljer, som skal være på plads for at sikre korrektheden af koden i sidste ende. Det kan ofte tage overraskende lang tid. Det kan betale sig at være omhyggelig med at lave korrekthedsanalyse af detaljerne (initialisering og opdatering af variable, f.eks.) i ens forslag til pseudo-kode, før man går videre til næste niveau. Hvis niveau 2 er lavet grundigt, bliver niveau 3 ofte relativt nemt.

Til forelæsningerne udtrykker vi os primært på niveau 1 og til dels på niveau 2. Lærebogen udtrykker sig også på niveau 1 og niveau 2, med de fleste algoritmer beskrevet meget detaljeret på niveau 2. I projektet arbejder vi med niveau 3, ofte baseret på niveau 2 fra bogen.

Til TE-timer er det i opgaver, hvor man skal udvikle, beskrive og/eller analysere algoritmer, nok at gøre dette på niveau 1, medmindre andet er angivet (dvs. medmindre der direkte bliver bedt om pseudo-kode eller Java-implementation).

Eksaminatorier I

1. Cormen et al. øvelse 7.1-1 (side 173).

2. Cormen et al. øvelse 7.1-2 (side 174). Her er q værdien som PARTITION returnerer, dvs. $(i + 1)$ i linie 8 i pseudo-koden i Cormen et al. side 171. Hint til det andet spørgsmål: Start med at lave princippet for det mørkegrå område på figur 7.2 (side 173) om fra " $> x$ " til " $\geq x$ " (overvej hvorfor dette ikke vil ændre på korrektheden af selve Quicksort), og fyld så det lysegrå og det mørkegrå område så balanceret som muligt under udførelsen af PARTITION.
3. Cormen et al. øvelse 7.2-2 (side 178). Antag her at det er PARTITION fra side 171, der bruges (ikke PARTITION varianten lavet i øvelse 7.1-2 ovenfor). Hint: i øvelse 7.1-2 (side 174), som er løst ovenfor, er der undersøgt i detaljer hvad PARTITION fra side 171 gør på det pågældende input. Argumenter for køretiden, når dette sker gentagne gange i Quicksort.
4. Cormen et al. øvelse 7.2-3 (side 178). Vis at det faktisk gælder både når input er stigende sorteret og når det er aftagende sorteret. Hint: undersøg i detaljer hvad PARTITION fra side 171 gør på de pågældende input. Argumenter så for køretiden, når dette sker gentagne gange i Quicksort.
5. Cormen et al. problem 7-2, spørgsmål b (side 186). Svar derefter igen på de to sidste øvelser [dvs. Cormen et al. øvelse 7.2-2 (side 178) og øvelse 7.2-3 (side 178)], under antagelse af at Quicksort bruger den nye PARTITION procedure (og laver to rekursive kald på de to dele af array A hvor PARTITION har placeret elementer forskellige fra $A[q]$).
6. Eksamen juni 2008, opgave 1 b. Man må gerne referere til oplysninger i bogen, når man giver begrundelser. Quicksort antages at bruge PARTITION fra side 171.
7. Cormen et al. øvelse 6.1-5 (side 154).
8. Cormen et al. øvelse 6.1-6 (side 154).
9. Eksamen juni 2008, opgave 4 a. Hob er en fordanskning af ordet heap.
10. Cormen et al. øvelse 6.1-4 (side 154).

Eksaminatorier II

I ugens anden (dobbel-)eksaminatorietime er de første 45 minutter afsat til at komme godt i gang med projektet, del I. Start med at arbejde på projektet senest dagen før, så I har konkrete spørgsmål med. Tanken er, at I tager laptop med og individuelt arbejder videre i timen, og undervejs kan få afklaret eventuelle spørgsmål, hvad enten I er kommet kort eller langt i projektet.

De sidste 45 minutter handler om nedenstående opgaver.

1. Eksamen januar 2008, opgave 1 b (sidehenvisningen skal være til side 164 i vores udgave (tredie) af Cormen et al.).
2. Eksamen januar 2006, opgave 1 b. Bemærk at der er tale om en min-heap.
3. Cormen et al. øvelse 6.2-1 (side 156).
4. Cormen et al. øvelse 6.5-9 (side 166).
5. (*) Bevis at de beregnede indexer i PARENT, LEFT og RIGHT på side 152 er korrekte (dvs. for en knude på arrayindex i altid giver index af dens forælder, venstre barn og højre barn). Hint: start med at vise det for knuder på stien mest til venstre i heapen.
6. (*) Cormen et al. problem 6.2 (side 167). Lav spørgsmål **e** før spørgsmål **d**. For spørgsmål **b**, brug f.eks. formel (A.5) side 1147.

Studiegrupper

Forslag til fokus for arbejde i studiegrupper (hvis man er i en sådan):

Diskuter hvad projektet går ud på, og hvordan I løser det. Forbered derudover dele af opgaverne til eksaminatorietimer, f.eks. på nedenstående måde.

- Lav eksamensopgaverne individuelt på forhånd, og ret hinandens i gruppen.
- Forsøg at løse opgaverne I.2–5 i fællesskab. Arbejd både med at få ideer på skitseplanet til de ønskede algoritmer og argumenter, og med at få dem formuleret præcist til sidst. I kan evt. dele disse opgaver

op imellem delgrupper, som senere forsøger at formidle de fundne løsninger til hinanden så klart og præcist som muligt.