Institut for Matematik og Datalogi
Syddansk Universitet, Odense

31. januar 2013
RF

DM507 – Opgaver uge 6

Eksaminatorier I

1. Cormen et al. øvelse 2.1-1 (side 22).

2. Implementer InsertionSort (side 18) i Java. Test at din kode fungerer ved at generere arrays med forskelligt indhold og sortere dem. [NB: Bogens pseudokode indekserer arrays startende med index 1, mens Java starter med index 0. Man kan evt. i Java bruge et array af længde $n+1$ og blot aldrig bruge index 0—så kan bogens pseudo-kode bruges direkte).]

   Tilføj tidtagning af din kode ved at indsættte to kald til metoden `System.currentTimeMillis()`, eet i starten af InsertionSort og eet i slutningen (slå funktionaliteten af metoden op i Javas online dokumentation). Der skal kun tages tid på selve sorteringen, ikke den del af programmet som genererer array'ets indhold.

   Kør din kode dels med sorteret input (best case for InsertionSort), dels med omvendt sorteret input (worst case for InsertionSort). Gør dette for mindst 5 forskellige værdier af n (antal elementer at sortere). Vælg værdier som får programmet til at bruge fra ca. 100 til ca. 5000 millisekunder (disse værdier er ikke de samme for best case og worst case). Gentag hver enkelt kørsel tre gange og find gennemsnittet af antal millisekunder brugt ved de tre kørsler (fluktuationer fra baggrundsprocesser får derved mindre indflydelse). Dividér de fremkomne tal med henholdsvis n (for best case input) og n^2 (for worst case input), og check derved hvor godt analysen passer med praksis – de resulterende tal burde ifølge analysen være konstante (for best case tallene og for worst case tallene, hver for sig), jvf. grafer på slides fra forelæsningen.

   Kør derefter din kode med input, som er random `int`'s (brug f.eks. `java.util.Random`, se evt. denne vejledning. Er køretiderne tættest på best case eller worst case?

3. Vis for $f(n) = 0.1 \cdot n^2 + 5 \cdot n^{1.5} + 5 \cdot n + 10 \cdot n^{0.5} + 25$ at $f(n) = \Theta(n^2)$ og $f = o(n^3)$. Hint: brug sætning fra slides om analyse af algoritmers køretider (står også i starten af opgave 8 nedenfor).

4. Vis at følgende funktioner er skrevet op efter stigende asymptotisk voksehastighed:

   $1,\ \log n,\ \sqrt{n},\ n/\log n,\ n,\ n\log n,\ n\sqrt{n},\ n^2,\ n^3,\ n^{10},\ 2^n$

   Mere præcist, vis for alle par $f(n)$ og $g(n)$ af naboer i listen at $f(n) = o(g(n))$. Hint: brug sætninger fra slides om analyse af algoritmers køretider (står også i opgave 8 nedenfor).

5. Cormen et al. øvelse 3.1-1 (side 52).

6. Cormen et al. øvelse 3.1-4 (side 53).

7. [Let udfordrende] Cormen et al. øvelse 3.2-3 (side 60). Hint: for at vise formel (3.19) kan man bruge Stirlings formel (3.18), sådan som bogen foreslår i tekst ved (3.19), men man kan argumentere mere jordnært ud fra definitionen af $n!$. Husk regnereglerne for logaritmer (se (3.15), specielt anden regel).

8. [Udfordrende] Bevis (ud fra definitioner af $\Theta()$ og $o()$ og af begrebet grænseværdi) sætningerne

   $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c > 0 \quad \Rightarrow \quad f(n) = \Theta(g(n))$$

   og

   $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \quad \Rightarrow \quad f(n) = o(g(n))$$

   fra slides om analyse af algoritmers køretider. Husk at definitionen af grænseværdi (jvf. MM501/MM529) er: $\lim_{n\to\infty} h(n) = c$ hvis der for ethvert lille tal $\epsilon > 0$ findes et N_0 så $|h(n) - c| \le \epsilon$ når $n \ge N_0$.

   Hvis der er tid: Bevis også sætningen

   $$\lim_{n\to\infty} \frac{n^a}{b^n} = 0 \text{ for alle } a > 0 \text{ og } b > 1$$

   (f.eks. ved gentagne anvendelser af l'Hôspitals regel, som angivet på slides), og repetér forelæsningens bevis for, hvordan nedenstående variant følger af ovenstående:

   $$\lim_{N\to\infty} \frac{(\log_c N)^a}{N^d} = 0 \text{ for alle } a, d > 0 \text{ og } c > 1.$$

Eksaminatorier II

1. Cormen et al. øvelse 2.1-3 (side 22). Lav blot pseudo-koden (ikke delen med invarianter, vi vender tilbage til begrebet invarianter senere i kurset).
2. Cormen et al. øvelse 2.2-3 (side 29). Svar også for best-case køretid.
3. Cormen et al. øvelse 2.3-5 (side 39).
4. Cormen et al. øvelse 2.3-1 (side 37).
5. Cormen et al. øvelse 2.3-2 (side 37).
6. Cormen et al. opgave 2-1 (side 39). Til spørgsmål d: tanken her er at bruge køretid i praksis til at fin-tune det endelige valg af k.
7. Cormen et al. opgave 2-4 (side 41). [Del **d** er udfordrende].
8. [Let udfordrende] Cormen et al. øvelse 2.3-7 (side 39). Hint: start med at sortere tallene med en $O(n \log n)$ sorteringsalgoritme.

Studiegrupper

Forslag til fokus for arbejde i studiegrupper:

Gennemgå afsnit 3.2 og diskutér hvilke af afsnittets ligninger og definitioner, I (alle eller blot nogle af jer) har mødt før i enten gymnasiet eller et kursus på Imada (f.eks. MM501, MM529, DM535).

Forbered dele af opgaverne til eksaminatorietimer, f.eks. på nedenstående måde.

- I opgave I.2, lav programmeringen og programkørsel (evt. i par) før gruppemøde. På mødet, sammenlign køretider (er jeres maskiner f.eks. lige hurtige?).
- Løs først opgave I.3 i fælleskab. Del jer derefter i to eller flere grupper, som tager hver sine dele (hver sine nabopar af funktioner) af opgave I.4. Forklar bagefter løsningerne/beregningerne for hinanden (alle bør forklare mindst én).

- Forsøg at løse de mere kreative og udfordrende af opgaverne i fælleskab (f.eks. I.5, I.6, I.7, I.8, II.6, II.7, II.8). 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 dise opgaver op imellem delgrupper, som senere forsøger at formidle de fundne løsninger til hinanden så klart og præcist som muligt.