

## Lab DM534

### Uge 46

Disse lab-timer handler om at undersøge og sammenligne forskellige algoritmer for sortering.

Hovedparten af tiden vil vi bruge på et program der kan udføre og visualiserer en række sorteringsalgoritmer. Programmet er skrevet af en tidligere Imada-studerende, Jacob Aae Mikkelsen.

Klik på linket til programmet på kursets webside. Har din browser rette Java-indstillinger kan den køre programmet direkte, ellers kan filen gemmes og køres i en shell med kommandoen `java -jar sorting.jar`.

Lav flg. opgaver. Arbejd gerne i par.

1. Afprøv forskellige valgmuligheder i programmet for at få en idé om hvad det tilbyder. Bemærk at tal i input repræsenteres af søjler, med en søjles højde svarende til tallets størrelse.
2. Vælg algoritmen *InsertionSort* og *Visualization of algorithms*. Start algoritmen, og udled hvad de forskellige farver på søjler angiver. Udled også hvad der menes med en "copy" i algoritmen. Det kan evt. være en fordel at ændre på hastigheden.
3. Gentag ovenstående, med forskellige typer input (random, decreasing, increasing). Hvad er værste og bedste antal sammenligninger, du kan få frem (for arraysize  $n$  lig 8, hvilket er default)? Svarer det til, hvad analyse giver os (afsnit 5.6 i bog)?
4. Skift til *Timetaking of the algorithms* og sæt arraysize til 10000 (under *Settings*). Kør alle 5 algoritmer for forskellige typer input. Hvilke ser ud til at være hurtigst? Er det ens for alle typer af input?
5. Kør alle algoritmer med random input, med inputstørrelser  $n$  lig 500, 5000, 50000 og 500000 (dog ikke hvis algoritmen tager mere end ca.

10 sekunder – i så fald afbrydes blot, og denne værdi af  $n$  undlades for den pågældende algoritme). Gentag hver kørsel tre gange og find gennemsnittet  $t$  af de tre gange. Plot i et koordinatsystem talpar  $(n, t)$ , dvs. med  $n$  på  $x$ -aksen og  $t$  på  $y$ -aksen. Hvis du kender et plot-program så brug dette, ellers skitser et plot på papir. [Hvis du bruger et program, og kan vælge logaritmiske skalaer på akser, så vælg dette på begge akser, det vil gøre plottet mere overskueligt.]

6. Vend tilbage til *Visualization of algorithms*, og følg med i algoritmerne. Kan du for *InsertionSort*, *SelectionSort* og *MergeSort* se at søjlernes bevægelser passer med algoritmens beskrivelse i bogen (henholdsvis side 225, 227 og 542–44)? Du kan evt. også kigge på koden vist til venstre i programmet.
7. Kan du finde et udtryk for, hvor mange sammenligninger *SelectionSort* bruger som funktion af  $n$ ? [Hint: tænk på analysen af *InsertionSort*.] Passer det med data fra kørsler af programmet?
8. Kan du udlede hvordan *Quicksort* fungerer (ud fra bevægelserne og/eller den viste kode)?