

Øving 3, algoritmer og datastrukturer

Sortering, sammenligne ulike typer quicksort

Oppgaven er å sammenligne vanlig quicksort som bruker ett delingstall, med varianten som bruker to delingstall (dual pivot quicksort).

Implementer begge sorteringsalgoritmene, som metoder i samme testprogram. Dere kan f.eks. bruke min quicksort fra læreboka, og dual pivot quicksort basert på <https://www.geeksforgeeks.org/dual-pivot-quicksort/>.

Bruk tidtaking, og finn ut hvilken sorteringsalgoritme som fungerer best på tabeller med 10–100 millioner elementer. Det er ingen fasit på dette, det vil avhenge av pc, programmeringsspråk og implementasjonsdetaljer.

Ta tiden på sortering av:

- Tabell med helt tilfeldige data
- Tabell med mange duplikater, f.eks. hvor annenhvert element er likt. Gode sorteringsalgoritmer er raskere på slike data, dårlige kan bli tregere.
- Tabell som er sortert fra før, da dette er et vanlig spesialtilfelle. Hvis det blir noen forskjell, bør dette gå raskere, ikke tregere.

De som vil ha ekstra utfordringer, må gjerne prøve ut ulike triks for å speede opp sorteringen enda mer. F.eks. trikset med å bruke innsettingssortering på tilstrekkelig små intervaller. (Hvor små slike intervaller bør være før man bruker innsettingssortering, er et interessant spørsmål i seg selv, det kan finnes med eksperimentering.)

Sjekk for korrekt sortering

Når sorteringen (og tidtaking) er ferdig, bruk en løkke som sjekker at tabellen er korrekt sortert. Altså at `tabell[i+1] >= tabell[i]` for alle `i`-verdier mellom 0 og `tabell.length-2`. Dermed finner dere ut om sorteringsalgoritmen deres virker som den skal. Det er fort gjort å gjøre feil. Man kan sortere *vel*di raskt, hvis resultatet ikke trenger å være riktig. :-)

Bruk en slik test etter hver av sorteringene, for å få oppgaven godkjent.

Tips oppgave 1

Tips ang. klasser i java/c++

En *klasse* er en *datatype*. Derfor gir vi klassene navn etter hva slags data de inneholder, og *ikke* etter hva slags algoritmer de inneholder. Algoritmene er *metoder*, så det er metodene som får navn etter algoritmene.

Når vi jobber med sortering, har altså *klassen* navn som «Talltabell» eller kanskje «Testprogram». *Metodene* har navn som «quicksort () », «min_sortering () », eller lignende.