

Øving 1, algoritmer og datastrukturer

Oppgaven gjøres og godkjennes på labben. Det er en fordel å forberede seg, f.eks. ved å lese gjennom oppgaven og gjøre unna enkle deler av programmeringen på forhånd.

Oppgave

Oppgave 1–1, 1–2 og 1–3 sidene 18–19 i læreboka.¹ Det fins mange løsninger på problemet, med kompleksitetene $O(n^2)$, $O(n)$ og $O(n^3)$. Programmer (minst) en av løsningene. Sjekk først at du får riktig svar ved å sammenligne med eksemplet i boka. Deretter skal du prøve ut kjøretiden ved å prøve forskjellige problemstørrelser. Hvilken kompleksitet har din algoritme? Sammenlign teoretisk kompleksitet med de praktiske målingene.

Noen tips

- Metoder som bare leter opp de to dagene med absolutt lavest og høyest kurs vil ikke virke. Dette fordi dagene kan komme i feil rekkefølge. Man må kjøpe aksjene *før* man selger dem.
- For å teste med store tabeller, kan dere f.eks. bruke `math.random` for å fylle tabellen med tilfeldige kursforandringer. Pass på at det blir både positive og negative tall. (Ikke ta tiden på initialiseringen, `math.random` er en treg metode.)
- En vanlig feil på denne oppgaven, er å programmere som om tabellen beskriver *verdien* på aksjene fra dag til dag. Det gjør den ikke, den beskriver hvordan verdien *forandrer seg*. Men det er ikke vanskelig å lage en løkke som regner ut hva aksje-verdien blir på dagene. Det er bare å starte med en passende startverdi, og legge til forandringene fra dag til dag. (Startverdi er ikke gitt, men påvirker heller ikke hva som lønner seg best.)
- I forelesningen så vi på triks for å få gode tidsmålinger selv om maskinklokka ikke er helt god.

¹ Bok: Hafting/Ljosland: *Algoritmer og datastrukturer*