

Ukeoppgave i uke 9, INF2440 – v2015

Oppgave 1

Det viser seg at det er tidsmessig tvilsomt å bruke Javas biblioteksklasse: `ArrayList <Integer>` for å lagre en mengde med heltall (noe du stadig trenger i `Oblig3`).

Prøv heller å lage en klasse `IntList` med samme metoder(interface) som du evt. ville bruke i `ArrayList`. For meg ble det: konstruktøren `IntList(int len)`, `add(int i)`, `get(int i)`, `clear()`, og `size()`. Der lagrer du heltallene direkte i en `int[] data`. Når du prøver å legge til ett ekstra heltall som det 'ikke' er plass til i arrayen, kan du f.eks lage en ny array som er dobbelt så stor som du nå har. Du kopierer så det gamle innholdet i 'data' over i første halvdel av den nye arrayen, og lagrer deretter det nye heltallet. Til sist flytter du array-pekeren 'data' til å peke på denne nye, dobbelt så store arrayen.

Mål tidene på f.eks. å lagre og så lese 1 mill. heltall i en `ArrayList <Integer>` mot det å gjøre det samme i et objekt av `IntArray`. Du vil sannsynligvis oppdage at din `IntList` går minst dobbelt så fort som en `ArrayList <Integer>`.

Oppgave 2

Det er viktig at du denne uka kommer opp med både en kompilerbar og kjørbart versjon av den sekvensielle delen av `Oblig3`. Jobb med det, og bruk din egen `IntList` istedenfor en `ArrayList <Integer>` hvis du greide å få `IntList` til å gå fortere.

Det er meget enkelt å bytte ut en klasse som en `ArrayList <Integer>` med `IntList` i et program, da det bare er i deklarasjonene at de er ulike; de har jo de samme metodene.