Gruppen består av Elias Daland, Endre Strand Norén, Kristoffer Albrigtsen og Marius Løvereide Borgen

d) Kjøretid for operasjonene i TabellMengde og LenketMengde

boolean inneholder(T element)

TabellMengde: O(n) – Må iterere gjennom tabellen for å finne elementet.

LenketMengde: O(n) – Må iterere gjennom nodene for å finne elementet.

JavaSetToMengde: O(1) gjennomsnitt – HashSet bruker en hashfunksjon, som gir O(1) for sjekking.

boolean erDelmengdeAv(MengdeADT<T> annenMengde)

TabellMengde: O(n * m), hvor n er størrelsen på den første mengden og m på den andre mengden. Må sjekke om hvert element finnes i den andre mengden.

LenketMengde: O(n * m) – Må traversere begge lenkene og sjekke for hver node.

JavaSetToMengde: O(m) – Bruker contains(), som i gjennomsnitt har O(1) for hver sjekk.

boolean erLik(MengdeADT<T> annenMengde)

TabellMengde: O(n), siden vi må sjekke elementene i begge mengder.

LenketMengde: O(n), iterasjon gjennom alle elementer for å sammenligne dem.

JavaSetToMengde: O(n) – Bruker equals() på HashSet, som sammenligner to sett i O(n) tid.

MengdeADT<T> union(MengdeADT<T> annenMengde)

TabellMengde: O(n + m) – Må iterere gjennom begge mengdene og legge til elementene.

LenketMengde: O(n + m) – Må iterere gjennom nodene og legge til elementene i unionen.

JavaSetToMengde: O(n + m) – Bruker add(), som har O(1) for hver operasjon, og begge mengder traverseres én gang.

T fjern(T element)

TabellMengde: O(n), må finne og fjerne elementet.

LenketMengde: O(n), iterasjon gjennom nodene for å fjerne elementet.

JavaSetToMengde: O(1) gjennomsnitt – remove() i HashSet har O(1) tid.

e) Analyse: HashSet vs TreeSet

Kjøretid for contains(T element)

HashSet: O(1) i gjennomsnitt for søk, da det benytter en hashtabell.

TreeSet: O(log n) for søk, da det benytter et balansert binærsøkstre.

Forskjell på HashSet og TreeSet:

HashSet er raskere for oppslag og innsetting (O(1) gjennomsnitt) men usortert.

TreeSet er langsommere for oppslag (O(log n)) men sorterer elementene. Dette kan være nyttig i tilfeller der sortering er viktig.

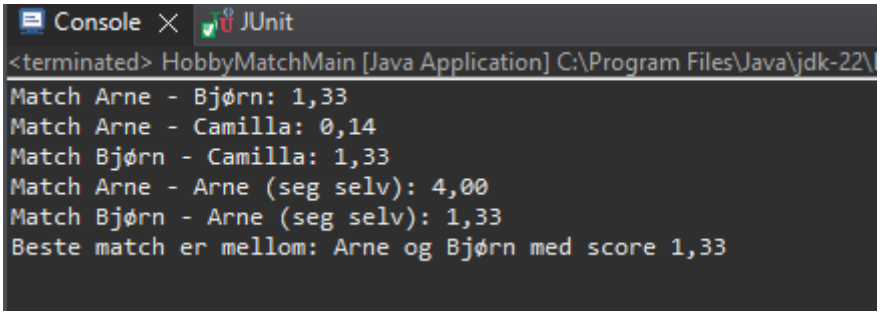
Forskjell på TabellMengde, LenketMengde og Set:

TabellMengde: O(n) for søk, men kan være raskere på små mengder.

LenketMengde: O(n) for søk, god plassutnyttelse ved dynamisk størrelse.

Set (HashSet): O(1) for oppslag i gjennomsnitt, men krever ekstra plass til hashtabellen.

Resultat av å kjøre HobbyMatchMain:



Opg 4 Uke 11;

Søk i HashSet: 10.000 tilfeldige tall ble søkt i HashSet. Tiden for søket ble målt til 13 ms.

Binærsøk i tabell: Etter at tabellen ble sortert, ble de samme 10.000 tallene søkt med binærsøk. Tiden for binærsøk var 11 ms.

HashSet har en gjennomsnittlig O(1) søketid per operasjon, så resultatet på 13 ms reflekterer hastigheten til HashSet.

Binærsøk har O(log n) kjøretid, og selv om binærsøk i en sortert tabell generelt er raskt, viste resultatet 11 ms, som er litt raskere enn søk i HashSet i dette tilfellet.

Konklusjon: Binærsøk var raskere enn HashSet i denne testen, noe som kan skyldes at binærsøk er mer effektivt i dette spesifikke tilfellet, til tross for at HashSet generelt gir raskere søk i gjennomsnitt.