IN2010
Innleveringsoppgave 0

Innlevering

Last opp filene dine på Devilry. Siden innleveringen ikke er obligatorisk, så gjelder denne fristen for å kunne få tilbakemelding. Merk at siden innleveringensoppgaven ikke er obligatorisk faller den heller ikke under reglementet for obligatoriske oppgaver; det innebærer at dere ikke har *krav* på tilbakemelding, men det er vår intensjon å gi alle som leverer tilbakemelding innen rimelig tid.

Vi anbefaler så mange som mulig om å samarbeide i små grupper på *opp til tre*. Dere må selv opprette grupper i Devilry, og levere som en gruppe (altså, ikke last opp individuelt hvis dere jobber som en gruppe).

For hver oppgave som ber om en implementasjon, skal dere levere kjørbar kode i Java eller Python. Vi anbefaler å skrive pseudokoden først og den kjørbare koden etterpå, og skrive ned eventuelle svakheter dere oppdaget da pseudokoden skulle oversettes til kjørbar kode.

Filene som skal leveres er:

- Én PDF som skal hete `IN2010-innlevering0.pdf`.
- Et kjørbart Java- eller Python-program for hver oppgave som ber om en implementasjon.

Filene skal ikke zippes eller lignende.

Merk at vi forventer riktig filtype. Filer som ikke er `.py`, `.java` eller `.pdf` vil ikke regnes som en del av innleveringen.

Oppgave 1: Ineffektive mengder

Den abstrakte datatypen for mengder kalles `Set`. Hvis `set` er av typen `Set`, så forventer vi at følgende operasjoner støttes:

`contains(set,` x`)`	er x med i mengden?
`insert(set,` x`)`	setter x inn i mengden (uten duplikater)
`remove(set,` x`)`	fjerner x fra mengden
`size(set)`	gir antall elementer i mengden

Operasjonene kan uttrykkes som metoder på objekter av klassen `Set` hvis man foretrekker det:

`set.contains(`x`)`	er x med i mengden?
`set.insert(`x`)`	setter x inn i mengden (uten duplikater)
`set.remove(`x`)`	fjerner x fra mengden
`set.size()`	gir antall elementer i mengden

Husk at hverken rekkefølge eller antall forekomster noen rolle i mengder. Ved fjerning av et element som ikke er i mengden skal mengden forbli uforandret.

Implementasjon

Vi ønsker *ineffektive* implementasjoner av alle operasjonene. Det er for å bygge intuisjon rundt kjøretid.

Alle operasjonene skal være implementert med *lineær tid*. Det vil si at hvis det er n elementer i mengden, så bruker hver operasjon $\mathcal{O}(n)$ steg i verste tilfelle.

Input

Input skal leses fra `stdin`.

Første linje av input består av et heltall N, der $1 \leq N \leq 10^6$, som angir hvor mange operasjoner som skal gjøres på mengden.

Hver av de neste N linjene er på følgende format

`contains` x	der x er et heltall $1 \leq N \leq 10^9$
`insert` x	der x er et heltall $1 \leq N \leq 10^9$
`remove` x	der x er et heltall $1 \leq N \leq 10^9$
`size`	

Merk at du ikke trenger å ta høyde for ugyldig input på noen som helst måte.

Output

Output skal skrives til `stdout`.

For hver linje av input som er på formen:

`contains` x

skal programmet skrive ut `true` dersom x er med i mengden, og `false` ellers.

For hver linje av input som er på formen:

`size`

skal programmet skrive ut antall elementer som er i mengden.

Eksempel input/output:

Eksempel-input	Eksempel-output
9	true
insert 1	false
insert 2	false
insert 3	2
insert 1	
contains 1	
contains 0	
remove 1	
contains 1	
size	

Det er publisert flere input- og outputfiler på semestersiden.

Oppgave 2: Binærsøk

I forelesningen ble det nevnt at datastrukturen kan påvirke kjøretiden på en algoritme. Gi et worst-case estimat av algoritmen nedenfor, som implementerer binærsøk over *lenkede lister*. Oppgi estimatet ved bruk av $\mathcal{O}$-notasjon. Hvordan påvirker valget av datastruktur kjøretidskompleksiteten i dette tilfellet?

ALGORITHM: BINÆRSØK MED LENKEDE LISTER

Input: En ordnet lenket liste A og et element x
Output: Hvis x er i listen A, returner **true** ellers **false**

```
1  Procedure BinarySearch(A, x)
2      low ← 0
3      high ← |A| − 1
4      while low ≤ high do
5          i ← ⌊(low+high)/2⌋
6          if A.get(i) = x then
7              return true
8          else if A.get(i) < x then
9              low ← i + 1
10         else if A.get(i) > x then
11             high ← i − 1
12     return false
```

Oppgi svaret ditt i PDF-en.