

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 innleveringsoppgaven 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 kjørbare 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 kjørbare kode.

Filene som skal leveres er:

- En 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`. Anta at `set` er av typen `Set`. Forventer vi at følgende operasjoner støttes:

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

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

<code>set.contains(x)</code>	er x med i mengden?
<code>set.insert(x)</code>	setter x inn i mengden (uten duplikater)
<code>set.remove(x)</code>	fjerner x fra mengden
<code>set.size()</code>	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 køen.

Hver av de neste N linjene er på følgende format

contains x	der x er et heltall $1 \leq x \leq 10^9$
insert x	der x er et heltall $1 \leq x \leq 10^9$
remove x	der x er et heltall $1 \leq x \leq 10^9$
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  $\leftarrow 0$ 
3   high  $\leftarrow |A| - 1$ 
4   while low  $\leq$  high do
5      $i \leftarrow \lfloor \frac{\text{low} + \text{high}}{2} \rfloor$ 
6     if  $A.get(i) = x$  then
7       return true
8     else if  $A.get(i) < x$  then
9       low  $\leftarrow i + 1$ 
10    else if  $A.get(i) > x$  then
11      high  $\leftarrow i - 1$ 
12  return false

```

Oppgi svaret ditt i PDF-en.