Innlevering 4

Høst 2023

Effektive mengder med hashing

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

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

Mengden skal implementeres *med hashing*. Det betyr at du må sørge for at `contains`, `insert` og `remove` er i (forventet amortisert) konstant tid. Operasjonen `size` bør være i $\mathcal{O}(1)$.

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 megnden.

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 insert 1 insert 2 insert 3 insert 1 contains 1 contains 0 remove 1 contains 1 size	true false false 2

Det er publisert flere input- og outputfiler på semestersiden.

Oppgaver

(a) Skriv et Java eller Python-program som leser input fra `stdin` og printer output *nøyaktig* slik som beskrevet ovenfor.

(b) Sammenlign tidsbruken på din implementasjon basert på hashing med din tidligere implementasjon basert på binære søketrær. Skriv en kort oppsummering av funnene.

Kattunge!

> Oppgaven er hentet fra Kattis[1]. Vi følger samme format på input- og output, slik at oppgaven deres kan lastes opp på Kattis, men dette er *ikke* et krav. Det er heller ikke nødvendig å oppfylle tidskravet som Kattis stiller.

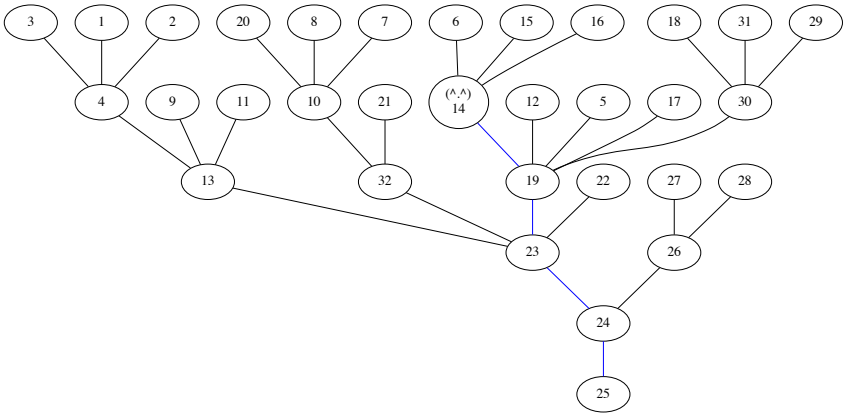
En kattunge sitter fast i et tre! Du må hjelpe med å finne ut hvordan den skal finne veien fra grenen den sitter på, og ned til roten av treet.

Input

Inputet beskriver et tre, der hver node kun inneholder et tall mellom 1 og 100.

- Første linje av input består av ett enkelt heltall K som angir noden hvor kattungen sitter fast.

[1] https://open.kattis.com/problems/kitten



Figur 1: Sti fra katten til roten

- De neste linjene består av to eller flere heltall $a, b_1, b_2, \ldots, b_n$, der a er foreldrenoden til nodene $b_1, b_2, \ldots, b_n$.
- Siste linje av input er alltid -1 som angir at treet er ferdig beskrevet.

Det er garantert at input beskriver *et tre*, altså er det garantert at hver node kun har én foreldrenode (det vil si at hver b_i kun forekommer ett sted i inputet).

Output

Oppgi stien fra der kattungen befinner seg til roten av treet.

Eksempel-input	Eksempel-output
14 25 24 4 3 1 2 13 9 4 11 10 20 8 7 32 10 21 23 13 19 32 22 19 12 5 14 17 30 14 6 15 16 30 18 31 29 24 23 26 26 27 28 -1	14 19 23 24 25

Det er publisert flere input- og outputfiler på semestersiden.

Oppgave

Skriv et Java eller Python-program som leser input fra `stdin` og printer output *nøyaktig* slik som beskrevet ovenfor.