

3 augusti Programmering, Pepper - Pass 6

1.

Implementera ditt eget träd från grunden. Du ska ha metoder för att lägga till noder, ta bort noder samt en metod som kollar ifall en nod existerar i trädet.

(a) Ta in en höjd på trädet från användaren, och konstruera ett perfekt balanserat träd med hjälp av den datastruktur som du nyss skapat.

(b) Skapa kod som visualiserar trädet.

2.

Givet funktionen

$$f(x) = (x^2 + x^3)/1000$$

Lös ekvationen $f(z) = 1$ med binary search. Det går lätt att använda numeriska metoder (t.ex. newton-raphson) men det är inte tillåtet för denna uppgift.

3. Teknologen Ture

Teknologen Ture har implementerat ett binärt sökträd. Trädet har följande krav:

- Om trädet inte är tomt existerar en rot.
- Varje nod i trädet har som mest två barn, ett till vänster och ett till höger.
- Varje nod innehåller en nyckel.
- För varje nod gäller att noder i delträdet åt vänster har strikt mindre nyckel än nodens nyckel och alla noder i delträdet till höger har strikt större nyckel än nodens nyckel.

När en nyckel sätts in i trädet skapas ett nytt löv i trädet med den givna nyckeln om nyckeln inte redan finns i trädet. Alla egenskaper ovan ska fortfarande uppfyllas. Om trädet är tomt så skapas en rot med den givna nyckeln.

I sitt träd ska Ture sätta in heltal mellan 0 och 1000 (inklusive). Ture har implementerat trädet och börjat sätta in heltal i det. När han har satt in några inser han att det kan vara så att trädet blivit obalanserat. Ture vill sätta in ett heltal till men han vill undvika att trädets höjd blir större än den redan är. Givet vilka heltal som satts in i trädet, hitta ett heltal i intervallet $[0,1000]$

som kan sättas in utan att trädets höjd ökar. Det nya heltal får inte redan finnas i trädet.

Tips: Notera att trädets höjd kan bli stor. Rekursiva funktioner som anropas på trädet kan nå gränsen för tillåtet rekursionsdjup i vissa språk. Speciellt i Python kan det ordnas genom att lägga till följande rader i programmet.

```
import sys
sys.setrecursionlimit(1500)
```

Input

En rad med n heltal, a_0, a_1, \dots, a_{n-1} : heltalen som Ture har satt in i sitt träd angivna i den ordning de sattes in. Det gäller att $1 \leq n \leq 1000, 0 \leq a_i \leq 1000$

Output

Ett heltal i intervallet $[0, 1000]$: en nyckel som går att sätta in i trädet utan att dess höjd ökar.

Om det finns flera lösningar, skriv ut vilken som helst. Om det inte finns någon lösning, skriv ut -1

Exempel

Nycklar som Ture satt `in` ? 4 2 8 7 10 0

Det går att sätta `in` nyckeln: 3

Nycklar som Ture satt `in` ? 11 5 17

Det går inte att sätta `in` nån nyckel: -1

Nycklar som Ture satt `in` ? 2 1 3 3

Det går inte att sätta `in` nån nyckel: -1
