Högskolan i Gävle

Binära sökträd och rekursiva metoder

Laboration 2

Ilayda Gül
Ilaayddaaaa@gmail.com

2025-04-22

Uppgift 1: Generics

1.Förklara varför kodraden boxTest(new Box); som återfinns långt ner på sidan 4 inte kommer att kompilera.

Svar: Raden **boxTest(new Box<Integer>);** Kompilerar inte eftersom den saknar konstruktorparenteser (). I java, när man ska skapa en instans av en generisk klass är den rätta syntaksen:

```
boxTest(new Box<Integer>());
```

Utan parantesen känner inte Java igen detta som ett uttryck för att skapa ett objekt.

2.Förklaring av BinarySearchTree<T extends Comparable<? super T>>

- T är typparametern
- extends Comparable<? super T> betyder att T måste kunna jämföras med sig själv eller en superklass till sig själv.

Detta villkor säkerställer att de element som läggs till i BST kan jämföras för att upprätthålla sorteringsordningen. Användningen av <? super T tillåter mer flexibla jämförelser och stödjer subtyp-polomorfism, vilket är särskilt användbart i generiska datastrukturer som ett binärt sökträd.

3. PECS Rule

PECS står för Producer Extends, Consumer Super. Det är ett minnestrick för att välja mellan extends och super när man använder wildcard i generics:

- Använd extends när en struktur producerar T-värden (endast läsning)
- Använd super när den konsumerar T-värden (endast skrivning)

I BSTs används oftast extends eftersom vi behöver läsa och jämföra element.

Uppgift 2: Binära sökträd

BinarySearchTree<T extends Comparable<? super T>> använder en inre klass BSTNode med attributen T item, BSTNode left och BSTNode right. BinarySearchTree behåller en referens till roten av trädet och en räknare för storleken.

De implementerade metoderna:

- Add(T item): Lägger till element i sorterad ordning
- searchFor(T item): Söker rekursivt efter ett element
- remove(T item): tar bort en nod genom att ersätta med in-order eftersöljare
- size(): returnerar antalet noder
- clear(): tar bort alla noder
- toString(): utför in-order traversal för att generera en sträng

Kollision hantering i add:

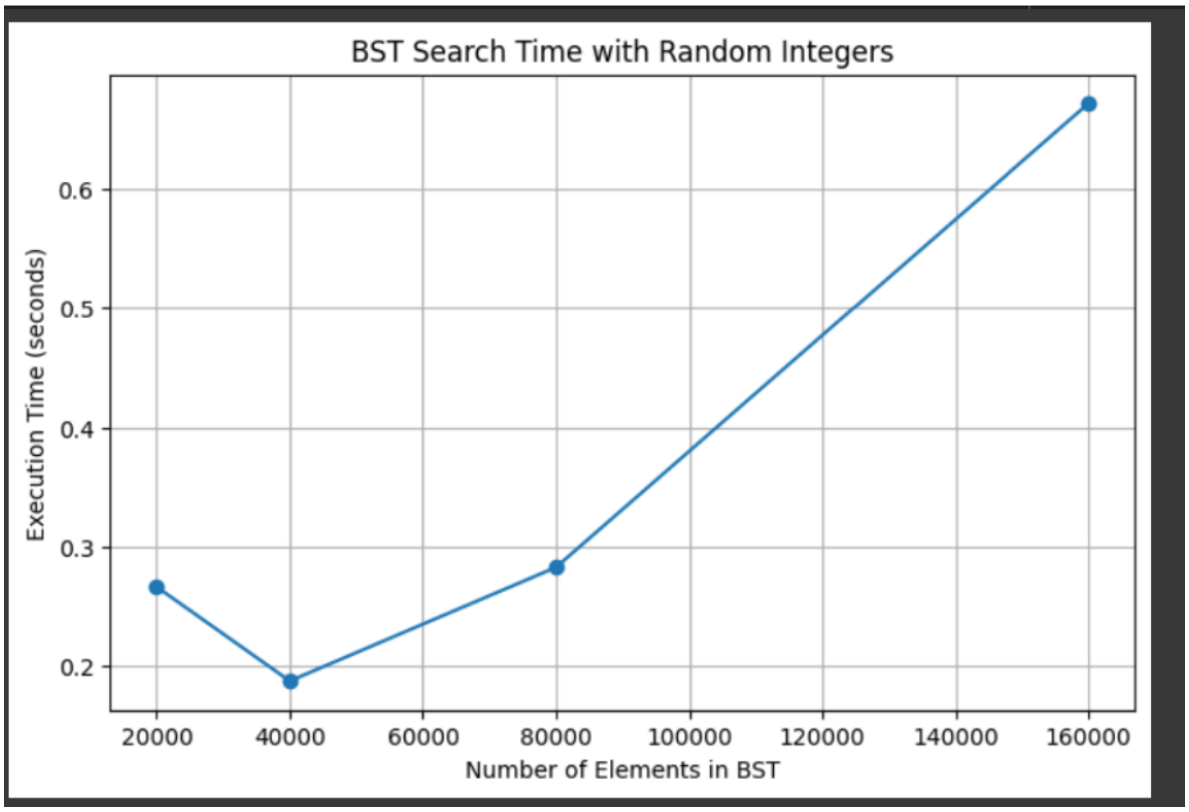
Ifall ett element med samma värde redan finns, ignoreras det. Detta val undviker dupliceringar och förenklar logiken för remove. Denna metod är lämplig när alla element måste vara unika, som i sökmotorer eller mängder.

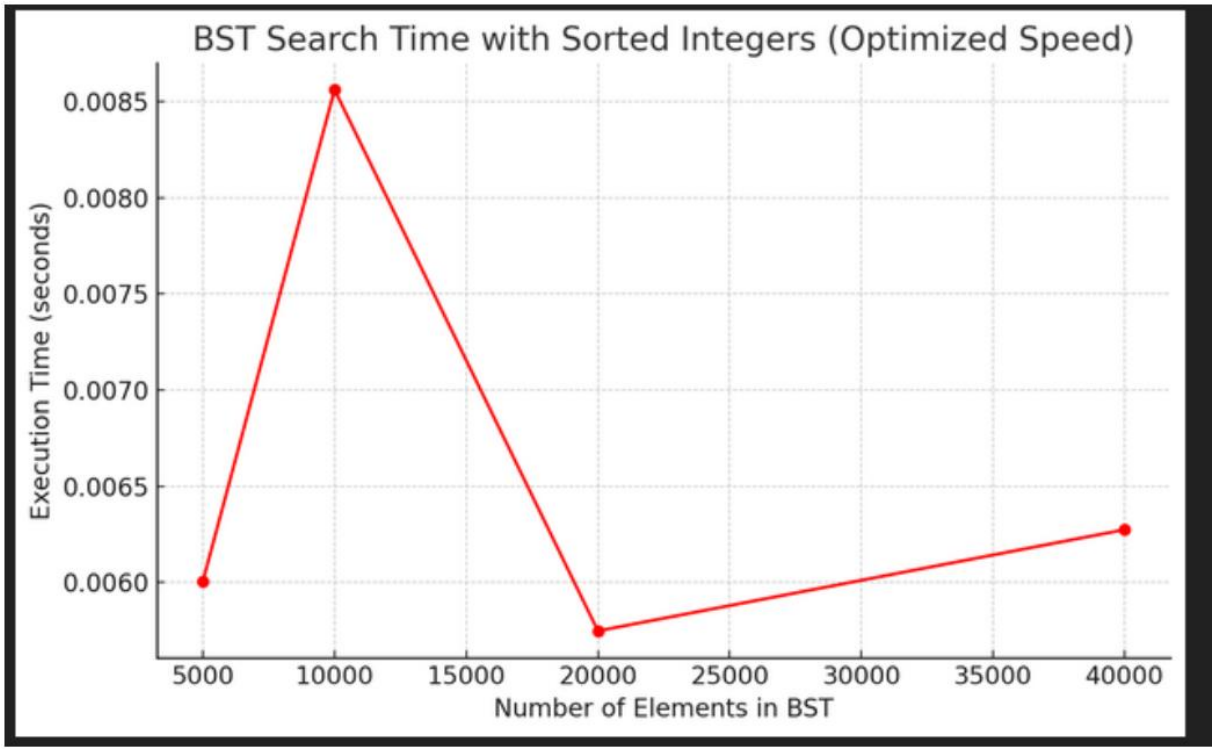
Påverkan på remove:

Eftersom dupliceringar inte är tillåtna, tar remove endast bort den första matchen som hittas vilket är unikt. Metoden förenklas till att hantera tre huvudfall:

- Noden har inga barn
- Noden har ett barn
- Noden har två barn - ersätt med in-order efterföljare.

Uppgift 3- Tidskomplexiteten:


BST Search Time with Random Integers
Execution Time (seconds)
0.2
0.3
0.4
0.5
0.6
20000
40000
60000
80000
100000
120000
140000
160000
Number of Elements in BST


BST Search Time with Sorted Integers (Optimized Speed)
Execution Time (seconds)
0.0060
0.0065
0.0070
0.0075
0.0080
0.0085
5000
10000
15000
20000
25000
30000
35000
40000
Number of Elements in BST