Duomenų struktūrų lab2 ataskaita

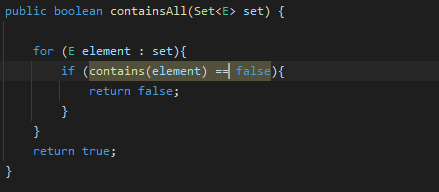
Parengė:

Vytenis Kriščiūnas IFF-1/1

**Tiriamieji metodai**

BstSet ir AvlSet containsAll() metodai.

Klasės BstSet: containsAll() metodas:



Metodas patikrina ar aibėje egzistuoja visi elementai, esantys aibėje set.

Klasė AvlSet paveldi containsAll() metodą iš BstSet klasės.

**Asimptotinis sudėtingumas**

Class BstSet: containsAll()

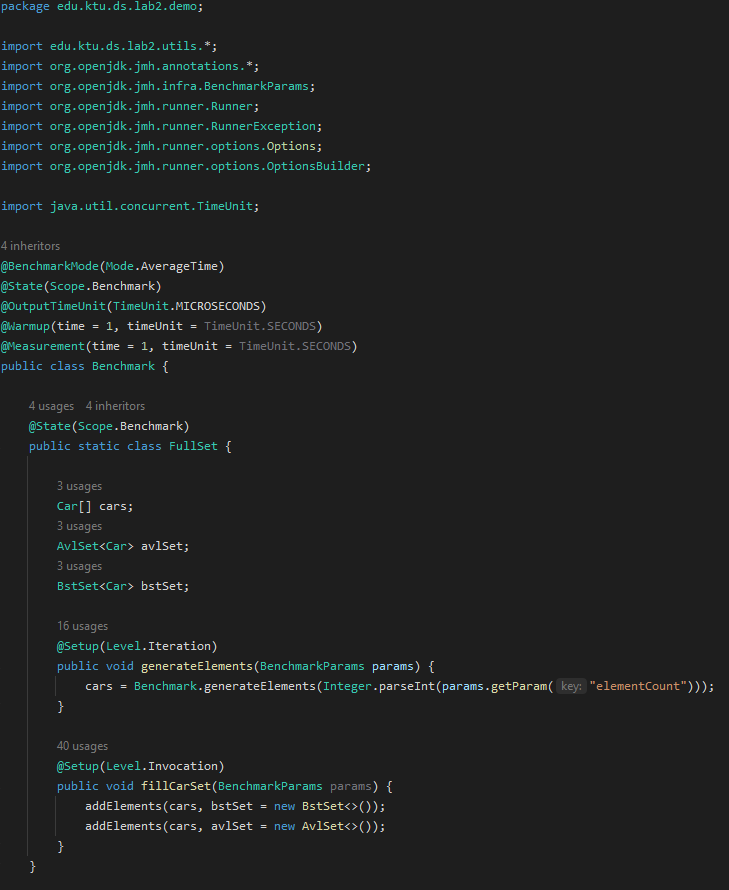
Asimptotinis sudėtingumas O(log2(n)).

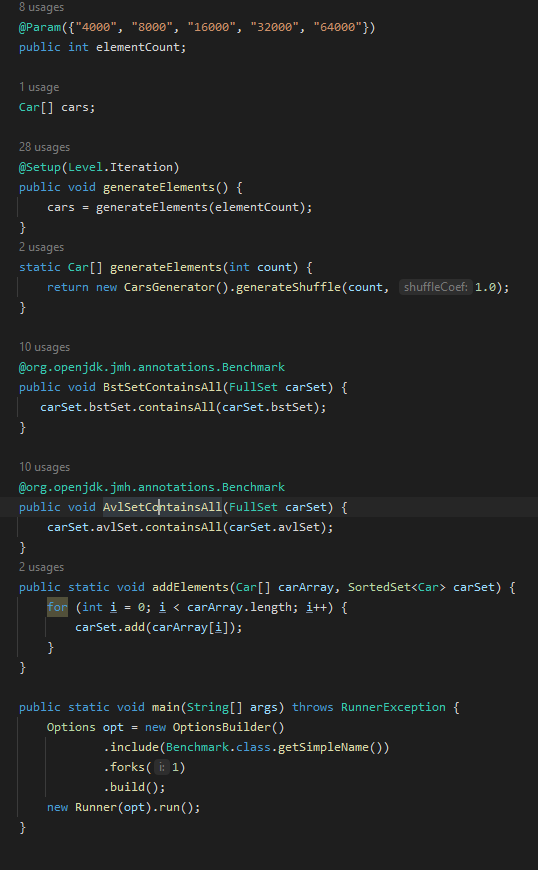
Class AvlSet: containsAll()

Asimptotinis sudėtingumas O(log2(n)).

**Greitaveikos testavimo metodika**

Testavimo klasė:





Pasirenku elementų kiekius testavimui. Sukuriu elementus pagal pasirinktą skaičių ir juos sudedu į BstSet ir AvlSet kintamuosius. Tada apsirašau metodus AvlSetContainsAll() ir BstSetContainsAll(), kuriuose naudoju containsAll() metodus. Galiausiai pradėjus greitaveikos testavimą stebiu gautus rezultatus.

Kompiuterio parametrai

Procesorius:

AMD FX-6300 six-core, greitis – 3.50 Ghz., 3 branduoliai, apdorojimas – 64-bit, talpykla – 8 MB.

Atmintis:

16 GB

Talpa:

SSD 224 GB

Algoritmų/ metodų vykdymo laiko priklausomybės nuo įvesties duomenų kiekio grafikas

Išvados

Pagal gautus rezultatus akivaizdu, kad AvlSet klasės containsAll() metodas yra greitesnis nei BstSet klasės containsAll() metodas, nors asimptotiniai sudėtingumai nesiskiria balansuoto medžio metodas turi atlikti mažiau veiksmų nei nebalansuoto medžio metodas. Kuo didesnis elementų kiekis tuo didesnis greičių skirtumas yra matomas.