

# L2 užduotis

## L2 privalomoji dalis

1. Išnagrinėkite Lab2\_AibesMedziai projekte pateiktą programinį kodą.
2. **(5 balai)** Realizuokite nerealizuotus programinio kodo metodus ir juos ištestuokite (testus galite inicijuoti tiek per projekto vartotojo sąsają, tiek per konsolę).

Klasėje BstSet reikia realizuoti šiuos metodus:

- remove(E element);
- removeRecursive(E element, BstNode<E> node) - pagalbinis remove metodas.
- Iteratoriaus metodą remove();
- addAll(Set<E> set) - patalpinti visus set elementus į aibę, jeigu abi aibės turi tą patį elementą jis nėra dedamas.
- containsAll(Set<E> set) - patikrinti ar aibėje egzistuoja visi elementai, esantys aibėje set.
- retainAll(Set<E> set) - išrinti elementus, kurių nėra aibėje set.
- headSet(E e) - grąžinamas aibės poaibis iki elemento e.
- tailSet(E e) - grąžinamas aibės poaibis nuo elemento e.
- subSet(E element1, E element2) - grąžinamas aibės poaibis nuo elemento element1 iki element2.

Klasėje AvlSet reikia realizuoti šiuos metodus:

- remove();
- removeRecursive(E element, AVLNode<E> node) - pagalbinis remove metodas.

3. **(3 balai)** Atlikite greitimeikos testavimą (naudokite JMH) su viena metodų pora iš žemiau pateiktos lentelės:

Variantas	Metodas 1	Metodas 2
1	Class BstSet: remove()	Class AvlSet: remove()
2	Class BstSet: contains()	Class AvlSet: contains()
3	Class BstSet: containsAll()	Class AvlSet: containsAll()
4	Class BstSet: addAll()	Class AvlSet: addAll()
5	Class BstSet: retainAll()	Class AvlSet: retainAll()
6	Class BstSet: remove()	Class java.util.TreeSet<E>: remove()
7	Class BstSet: contains()	Class java.util.TreeSet<E>: contains()
8	Class AvlSet: remove()	Class java.util.TreeSet<E>: remove()
9	Class AvlSet: contains()	Class java.util.TreeSet<E>: contains()

**Pastaba: Konkretų variantą gausite iš laboratorinių darbų dėstytojo.**

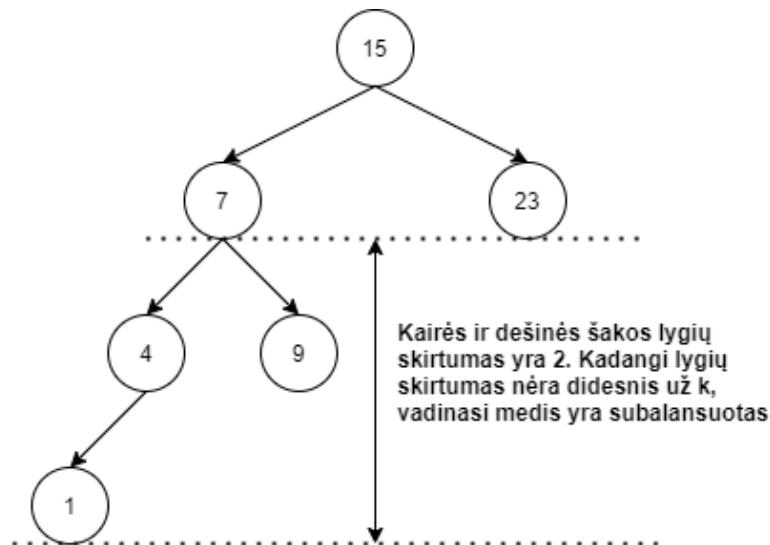
Laisva forma parenkite ataskaitą. Ataskaitoje būtinai turi būti:

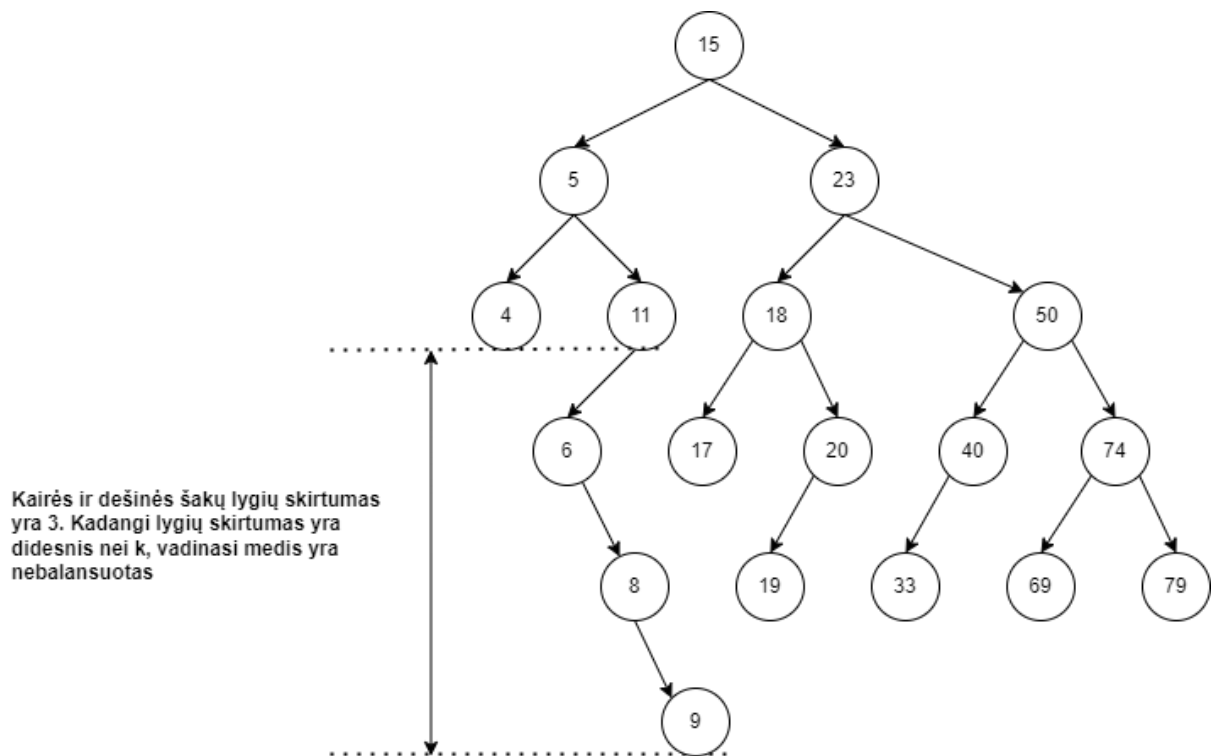
- Aprašyti tiriamieji metodai.
- Pateiktas tiriamųjų metodų asimptotinis sudėtingumas.
- Aprašyta greیتaveikos testavimo metodika (testavimo algoritmas).
- Aprašomi kompiuterio, su kuriuo buvo atlikti greیتaveikos testai, pagrindiniai parametrai (procesoriaus charakteristikos, atminties kiekis ir pan.).
- Algoritmų/metodų vykdymo laiko priklausomybės nuo įvesties duomenų kiekio grafikas.
- Išvados. Išvadose laisva forma turėtų būti pateikti atsakymai į šiuos klausimus:
  - Ar eksperimentiškai nustatyta vykdymo laiko priklausomybė nuo įvesties dydžio atitinka teorinį algoritmo/metodo asimptotinį sudėtingumą. Jei ne, kodėl?
  - Kurie iš tirtųjų metodų vykdymo laiko požiūriu yra geresni. Kodėl?

## L2 neprivalomoji dalis

Išspręskite šiuos uždavinius:

1. (1 **balas**) Turime dvejetainį paieškos medį su jūsų pasirinktais duomenimis (medyje saugomas objektas turi realizuoti sąsają Comparable). Patikrinkite, ar medžio šakų balansas neviršija sveiką skaičiaus  $k$  ( $k \geq 0$ ). Pavyzdys, jeigu  $k = 2$ :



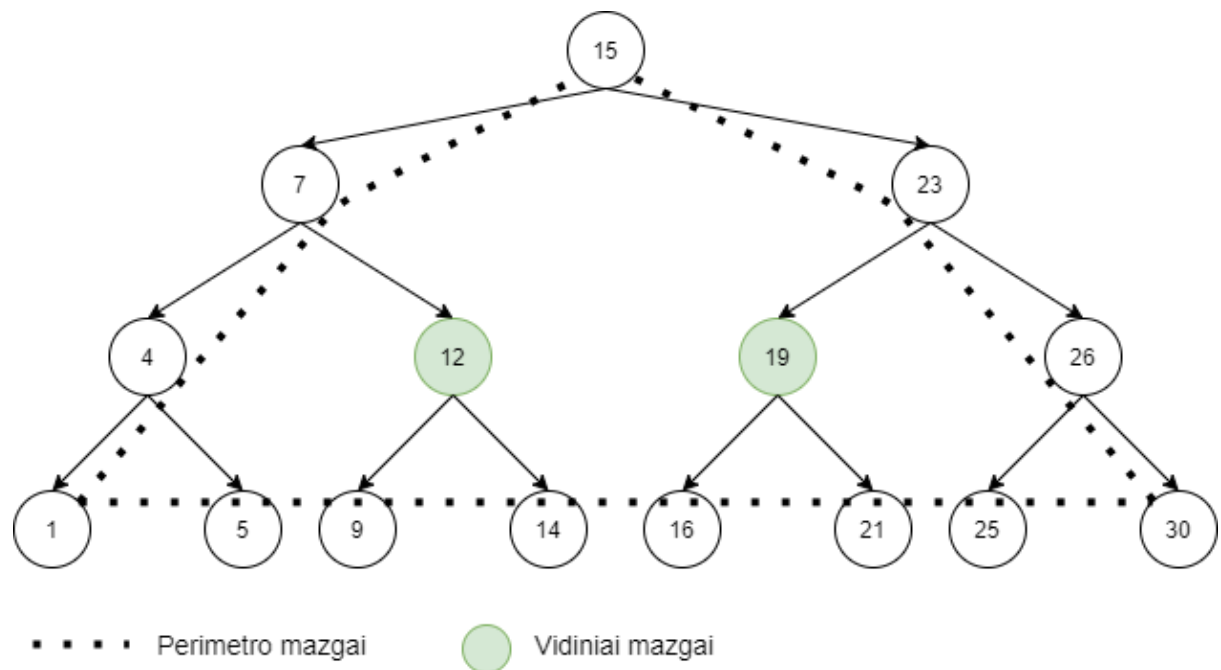


Ivesties duomenys:

- Skaičius  $k$
- Dvejėtinis paieškos medis

Parašykite algoritmą, kuris nustatytų, ar duotasis paieškos medis yra subalansuotas pagal parametą  $k$ . Algoritmo asimptotinis sudėtingumas turi būti  $O(n)$ .

2. (1 **balas**) Turime pilnąjį dvejėtinį medį. Parašykite algoritmą, kuris rastų medžio vidinius mazgus, t.y. mazgus, esančius medžio perimetro viduje. Pavyzdys:



## LD2 gynimas

Gynimo metu dėstytojai gali studentų prašyti atlikti žemiau pateiktas užduotis ir atsakyti į žemiau pateiktus klausimus. Klausimų ir užduočių sąrašas nėra baigtinis - dėstytojai gynimo metu gali užduoti su tema susijusių klausimų ar praktinių užduočių, kurių nėra šiame sąrašė.

### Galimos praktinės užduotys iš privalomosios dalies

Realizuokite vieną ar kelis metodus iš šio sąrašo:

Gražinama reikšmė	Metodo aprašymas
boolean	<code>removeAll(BstSet&lt;?&gt; c)</code> Removes from this set all of its elements that are contained in the specified collection.
E	<code>ceiling(E e)</code> Returns the least element in this set greater than or equal to the given element, or null if there is no such element.
E	<code>floor(E e)</code> Returns the greatest element in this set less than or equal to the given element, or null if there is no such element.
E	<code>higher(E e)</code> Returns the least element in this set strictly greater than the given element, or null if there is no such element.
E	<code>last()</code> Returns the last (highest) element currently in this set.
E	<code>lower(E e)</code> Returns the greatest element in this set strictly less than the given element, or null if there is no such element.
E	<code>pollFirst()</code> Retrieves and removes the first (lowest) element, or returns null if this set is empty.
E	<code>pollLast()</code> Retrieves and removes the last (highest) element, or returns null if this set is empty.
<code>SortedSet&lt;E&gt;</code>	<code>copyOf(BstSet&lt;? extends E&gt; set)</code> Returns a Set containing the elements of the given set

### Galimos praktinės užduotys iš neprivalomosios dalies

1. Parašykite algoritmą, kuris patikrintų, ar duotasis dvejetainis medis yra dvejetainis paieškos medis.
2. Parašykite metodą, kuris konvertuotų masyvą į balansuotą dvejetainį paieškos medį.

3. Parašykite metodą, kuris konvertuotų dvejtainį paieškos medį į vienkryptį susietąjį sąrašą.
4. Parašykite metodą, kuris leistų iteruoti per medį išilgai jo lygių.
5. Parašykite metodą, kuris leistų iteruoti per medį vertikaliais sluoksniais nuo viršaus į apačią.
6. Išspausdinkite dvejtainio paieškos medžio lapus.
7. Duoti du medžio mazgai, raskite ir išspausdinkite trumpiausią kelią tarp šių mazgų.
8. Raskite ir išspausdinkite visus dvejtainio paieškos medžio kelius nuo šaknies iki lapo.
9. Duota skaičius, nustatykite, ar dvejtainiame paieškos medyje egzistuoja toks kelias, kurio mazgų raktų suma yra lygi duotajam skaičiui.
10. Perrašykite vieną iš pagrindinių medžio operacijų nenaudodami rekursijos.

## Galimi teoriniai klausimai

1. Koks yra įvairių laboratoriniame darbe realizuotų operacijų asimptotinis sudėtingumas?
2. Kokie yra dvejtainio paieškos medžio bei balansuoto dvejtainio paieškos medžio privalumai ir trūkumai?
3. Mokėkite paaiškinti įvairius su programiniu kodu susijusius niuansus.