

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Duomenų struktūros (P175B014)

Žodžių paieška naudojant 2-3 medį.

Atliko:

IFF 8/9 gr. studentas Benas

Miliūnas Priėmė:

Eimutis

Karčiauskas

KAUNAS

2019

1. Užduotis 2-3 Tree

Realizuoti 2-3 medžio contains, visitInOrder, insert metodus ir juos panaudoti žodžių paieškai su 370000 failo dydžiu.

Sukurti grafinę sąsają, kurioje būtų rodoma visų žodžių skaičius, kiek laiko užtruko paieškos operacija, ir viso rastų žodžių.

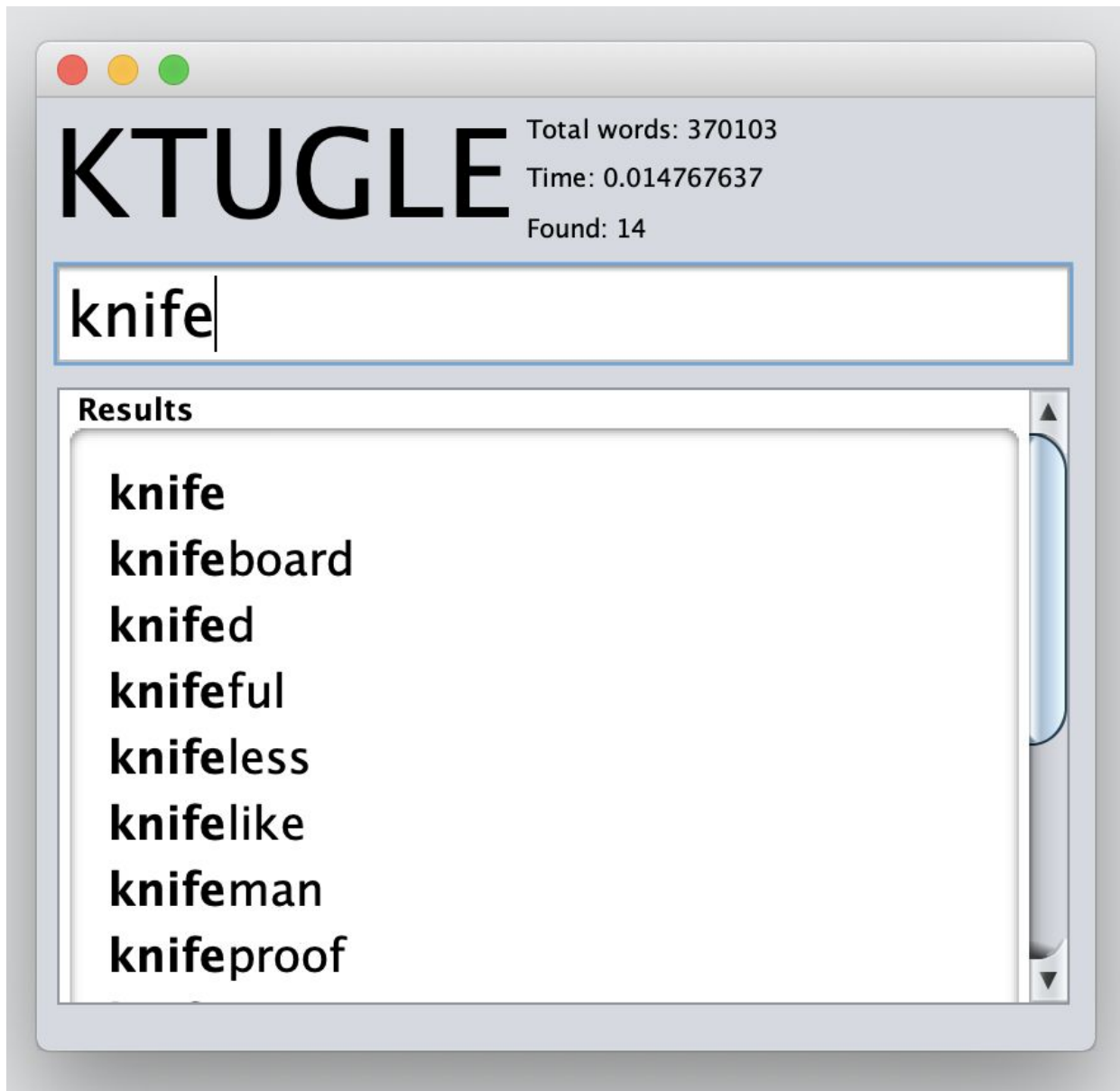
2. Užduoties analizė

Paleidžiant programą žodžius iš failo sudėsime į ką tik sukurtą 2-3 paieškos medį.

Į laukelį įvesime žodį, kurio norime ieškoti ir rezultatų lange matysime visus žodžius, kurių pradžia atitinka su ieškomu



3 . Testavimo pavyzdžiai





KTUGLE

Total words: 370103

Time: 0.013564967

Found: 5

Results

hello

helloed

helloes

helloing

hellos



KTUGLE

Total words: 370103

Time: 7.297E-6

Found: 0

4. 2-3 Medžio veikimas

Paieška

Pradiniai atvejai:

1. Jeigu T tuščias, return False .
2. Jei dabartinis node turi reikšmę, kurios ieškome, return True.
3. Jeigu aplookėme viską ir neradome ieškomos reikšmės, return False.

Recursive Calls:

1. Jei $K < \text{currentNode.leftVal}$, ieškome toliau kairėje medžio pusėje.
2. Jei $\text{currentNode.leftVal} < K < \text{currentNode.rightVal}$, ieškome toliau dešinėje medžio pusėje.
3. Jei $K > \text{currentNode.rightVal}$, ieškome dešinėje medžio pusėje

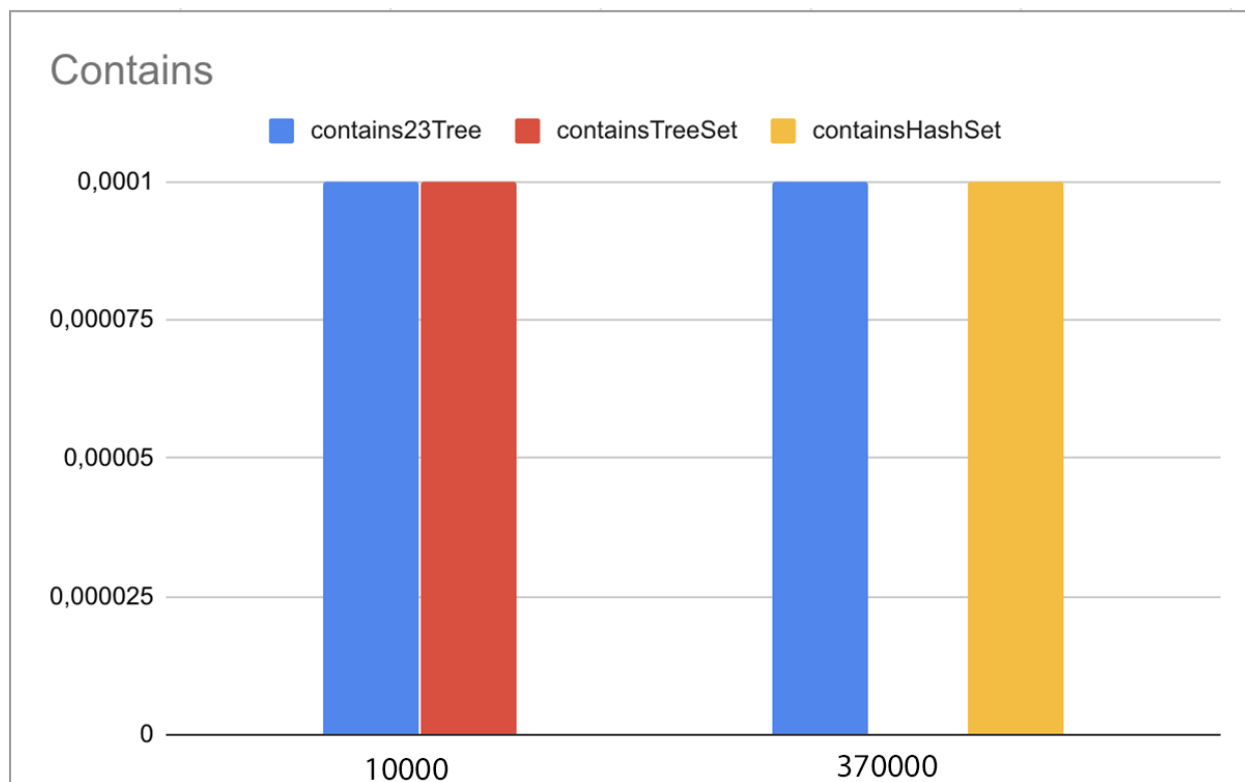
Įterpimas

Įterpimas prasideda suradus tinkamą įterpimo vietą. Jeigu po įterpimo mazgas turi keturis kaimynus, tada mazgas dalinamas į dvi lygias dalis, ir vidurinysis mazgas yra paverčiamas tėvu. Procesas kartojamas, kol pasiekiame tokį tėvinį mazgą, kuris turi tik du vaikus ir jo nereikia dalinti pusiau arba jeigu pasiekiame *root* viršūnę. Su šiuo algoritmu operacijų kiekis yra proporcingas aukščio medžiui. Visi lapai yra tame

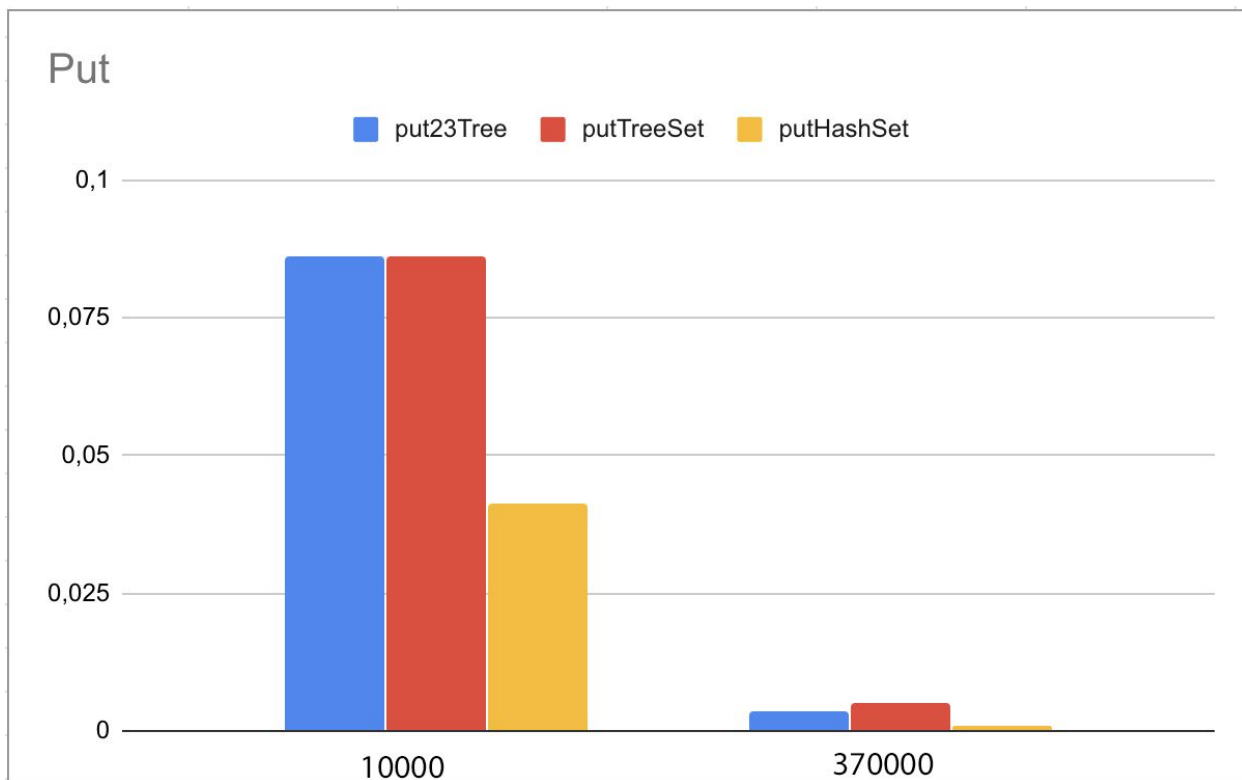
pačiame gylyje.

5. Greitaveika

Contains: 23Tree, TreeSet, HashSet



Insert: 23Tree, TreeSet, HashSet



6. Išvados

1) Paieška AVL ir 2-3 medyje yra $O(\log n)$, bet balansavimas AVL medyje yra $O(\log n)$ o , 2-3 medyje - $O(1)$. So, in 2-3 trees you have faster inserts at the expense of slower searches (since height is more compared to AVL trees).

2) 2-3 medį pasirinkau todėl, nes buvo įdomu realizuoti Avl ir Bst medžius.

3) Tikėjau, kad 2-3 medžio realizacija bus daug lengvesnė, tačiau vien įterpimui ir medžio balansavimui įsisavinti bei realizuoti prireikė apie 20 valandų. Pačio projekto trukmė - apie 40 valandų.

4) Iš greیتaveikos tyrimų galime matyti, kad mano realizuotas 2-3 medis veikia panašiu greičiu, kaip ir javos TreeSet, tačiau pranašiausiai veikia HashSet medis.