ADATSZERKEZETEK ÉS ALGORITMUSOK

AVL

Órai feladat 1

Nyisd meg a kiadott kódot és implementáld a balra forgatást!

`avl_tree<T>::_rotate_left`

Órai feladat 2

Nyisd meg a kiadott kódot és implementáld a jobbra forgatást!

`avl_tree<T>::_rotate_right`

Órai feladat 3

Nyisd meg a kiadott kódot és implementáld a kiegyensúlyozó függvényt!

`avl_tree<T>::_rebalance`

Több template típusparaméter (ismétlés)

Több template típusparamétert egyszerűen felsorolással csinálhatunk:

```
template<class K, class V>
class avl_tree
{ //... };

template<class K, class V>
const V & avl_tree<K, V>::iterator::value() const
{
      //...
      return n->value;
}
```

Gyakorló feladat G07F01

Alakítsátok át úgy az AVL fát, hogy kulcs-érték párokat lehessen benne tárolni. Második template paraméterként várja az érték típusát. A kulcshoz tartozó értéket a node-ban tárolja.

Példa a használatára:

```
avl_tree<int, string> szamok;
szamok.insert(6, “hat”);
avl_tree<int, string>::iterator it = szamok.find_it(6);
cout << “6 = “ << it.value();
```

Gyakorló feladat G07F01 (folytatás)

Az így megírt map segítségével

- hozzatok létre egy telefonkönyvet,
- szúrjatok bele pár név/telefonszám párost,
- keressetek rá egy-egy név telefonszámára,
- majd pedig iteráljatok végig a telefonkönyvön és írjátok ki az elemeit a konzolra.

Gyakorló feladat G07F02

AVL fákat, mint halmazokat felhasználva valósítsd meg az unió, metszet és különbség műveleteket!

Ehhez tölts fel két AVL fát véletlen számokkal, majd végezd el a műveleteket!

Művelet: Két AVL fa paraméter, és egy AVL fa visszatérési érték! (Vagyis valóban művelet!)

(megj.: Mivel két egyforma csúcs nem lehet ezért mondhatjuk, hogy halmazok!)

Gyakorló feladat G07F03

AVL fák segítségével készíts egy angol-magyar illetve egy magyar-angol szótár adatbázist.

A fordítandó szavai legyenek a csomópontok kulcsai és a hozzájuk tartozó fordítás pedig az értékek.

- Legyen lehetőség arra, hogy akár több hasonló jelentésű fordítást is felvegyünk egy adott szóhoz.

Készíts felhasználóbarát menüt, ami alkalmas a szótárak használatára:

- Tudjunk keresni egy adott szót és kiírni a hozzá tartozó fordításokat.
- Ha a keresett szó nem szerepel a fában, akkor vegyük fel legalább egy fordítással.

Gyakorló feladat G07F04

Készíts valamilyen iterátort a fához! Egyéni választásod szerint lehet külön iterátor osztály, amelyből több példány is lehet, vagy egyetlen act mutató az AVL fa tagjaként, ahogyan a láncolt listánál láttuk.

Minimális műveletkészlet az iterátornak:

- Fa legkisebb elemére állítás
- Rákövetkező elemre állítás
- Mutatott csúcs értékének lekérdezése
- Lekérdezni, hogy túlfutottunk-e az utolsó elemen

Gyakorló feladat G07F05

Generáljatok kellően sok véletlen adatot, majd szúrjátok be egy AVL fába. (Figyeljünk arra, hogy az AVL fa kiszűri az ismétlődéseket! Ha véletlenül egy már létező elemet hoztunk létre, akkor generáljunk helyette újat!)

Készítsetek statisztikát, hogy mennyi időbe telt egy elem átlagos beszúrása, összesen mennyi idő kellett az AVL fa felépítéséhez, hány forgatásra volt szükség a kiegyensúlyozásokhoz.

Ha elkészült az AVL fa akkor olvassátok ki az elemeket sorrendben egy vektorba.

A most már rendezett elemeket ismét szúrjátok be egy új AVL fába és készítsétek el ugyan azt a statisztikát, mint első esetben.

Értékeljétek ki (txt-ben) a tapasztaltakat!

Gyakorló feladat G07F06

Hasonlítsa össze a láncolt lista, bináris fa és AVL fa futási idejét nagy elemszámú (100 ezer), kis elemszámú (1000) beszúrás, törlés esetén.

A main függvény elején vonja le a következtetéseket. (5-7 mondat)