

# ADATSZERKEZETEK ÉS ALGORITMUSOK

Piros-Fekete fa

# Órai feladat

Előadáson tanult esetek szerint, illetve a gyakorlati videók alapján valósítsd meg a piros-fekete fa osztály `_rebalance_after_insert` és `_rebalance_after_remove` függvényét!

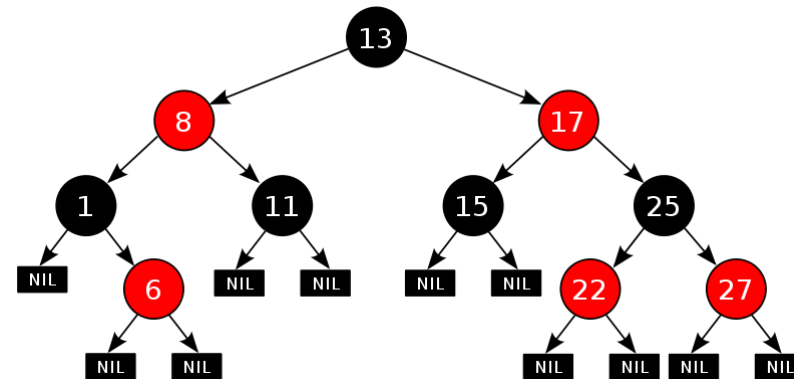
# Gyakorló feladat G07F01

Készíts vizualizációt a piros-fekete fához! Látható legyen a csúcsok kulcsa, színe, illetve a szülő-gyerek kapcsolatok a fában!

Nem muszáj grafikus megjelenítést csinálni, jó a konzolos is. Akár el is lehet forgatni a fát. Példa:

```

                                (27)
                                [25]-+
                                |    (22)
                                (17)-+
                                |    [15]
                                [13]-+
                                |    [11]
                                ( 8)-+
                                |    ( 6)
                                [ 1]-+
```



Bal oldalt a fekete csúcsok szögletes, míg a pirosak kerek zárójelben láthatók.

# Gyakorló feladat G07F02

Készíts egy teszt programot, amely összehasonlítja a Bináris Kereső, az AVL illetve PF fa használatát!

Ugyanazokat az elemeket (milliós nagyságrendű darabszám) szúrd be a BinKer, az AVL illetve a PF fába is és nézd meg, hogy hogyan változnak a fák magasságai (ezer, tízezer, százezer, millió ... elem után).

A következő tesztek tudja elvégezni a program:

Elemek növekvő/csökkenő sorrendben történő beszúrása

Random elemek beszúrása

# Gyakorló feladat G07F03

A **map** egy rendezett asszociatív konténer, mely kulcs-érték párokat tárol, ahol a kulcsok egyediek.

A feladat egy ilyen **map** osztály implementálása a Piros-fekete fa segítségével.

Legyen lehetőség a **map** osztálynál

- új elemet felvenni(beszúrás)
- elemet törölni
- Adott elemet(kulcs) lekérdezni
- kilistázni a tartalmát

A kulcs és az érték, lehet integer, string, és character is. (Template használata)

# Gyakorló feladat G07F04

- Egy map segítségével készítsenek statisztikát egy nagyon hosszú ékezetek nélküli (ASCII) szöveg szavainak előfordulására.
- **Bemenet:**
  - a program bemeneteként egy "in.txt"-t fogadjon, amely a feldolgozandó szöveget tartalmazza.
  - a szövegben található írásjegyeket ne vegyék figyelembe, továbbá kis és nagy betű sem számít.
- **Kimenet:**
  - **a)** írássátok ki a szavakat ábécé rendben (mellé az előfordulásuk számát)
  - **b)** írássátok ki a szavakat az előfordulásuk számának **csökkenő** sorrendjében, ha több szám ugyanannyiszor szerepelt, akkor ezeket soroltassátok fel a szám mellett (lásd pl.)

# Gyakorló feladat G07F04

- **Segítség:** ha az órán elkészített piros-fekete fa node osztályába egy kulcs-érték párost helyeztek és a kulcs szerint rendeztek, akkor egy map adatstruktúrát kaptok
- **Példaszöveg:**  
„That can I; At least, the whisper goes so. Our last king,  
Whose image even but now appear'd to us, Was, as you  
know, by Fortinbras of Norway, Thereto prick'd on by a  
most emulate pride, Dared to the combat; in which our  
valiant Hamlet-- For so this side of our known world  
esteem'd him-- Did slay this Fortinbras; who by a seal'd  
compact, Well ratified by law and heraldry, (...)”

# Gyakorló feladat G07F04

## a) elvárt kimenet:

a ---> 5  
against ---> 1  
all ---> 1  
and ---> 9  
appear ---> 2  
article ---> 1  
as ---> 3  
at ---> 1  
been ---> 1  
but ---> 2  
by ---> 8  
can ---> 1  
carriage ---> 1  
(...)



# Gyakorló feladat G07F04

## **b) elvárt kimenet**

12 --> of, the

9 --> and

8 --> by

7 --> d, our, to

5 --> a, this, which

4 --> fortinbras, his, in

3 --> as, so

2 --> appear, but, did, for, had, hamlet, hath, he, i, is, it, king, lands, norway, now, that, those, us, was, well

1 --> against, all, article, at, been, can, carriage, chief, combat, compact, competent, compulsory, conqueror, covenant, dared, design, diet, doth, emulate, enterprise, esteem, even, father, fell, food, foresaid, forfeit, full, gaged, goes, hand, haste, head, heraldry, here (...)

# Gyakorló feladat G07F05

Programozd le egy színes pónikat árusító játékbolt működését!

A póniknak két jellemző tulajdonsága van: a méretük (kicsi, közepes, nagy) és a színük. A színek fajtái és ezek sorrendje a következő:

- piros < barackszín < aransárga < ekrü < aloevera < zöld < akvamarin < cián < tengerészkék < ametisztlila < padlizsán < fekete

A játékboltba új színes póni készlet érkezett. Töltsd fel a raktárt a megérkezett játékokkal!

- A raktárkészletet piros-fekete fával reprezentáld!
- A játékokat a raktárban szín szerint rendezik, az azonos színeken belül pedig méret szerint.
- A raktárkészletet 100 új játékból áll. Ezeket véletlenszerűen generáld (random mérettel és színnel), majd szúrd be a fába!
- A raktárkészletbe több ugyanolyan póni is érkezhet, ezeket "egymás mögé" tesszük.
  - Ehhez a reprezentáló piros-fekete fa Node osztályát egészítsd ki úgy, hogy azt is tárolja, hogy az adott fajtájú játékból hány darab van!
- Írd ki a raktár tartalmát a rendezett sorrendben!
  - Pl.: piros póni: 12 db, barackszín póni: 23 db, ...

# Gyakorló feladat G07F05

A játékboltba a nap folyamán 200 vevő érkezik. Mindegyikük egy darab pónit szeretne venni. Szimuláld a vevőket és a vásárlásokat!

Egy vásárlás folyamata a következő:

- A vevő megérkezik, szeretne venni egy véletlenszerűen választott méretű és színű pónit.
- Az eladó ellenőrzi a raktárban, hogy az adott póniból van-e még.
  - Ha igen, eladja (a raktárban a számukat eggyel csökkenti)
  - Ha nem, akkor sajnos nem tudja kiszolgálni a vevőt
- Amikor egy pónifajta elfogy, töröld a csúcsát a fában!
- A szimuláció során a vásárlások részleteit írd ki a konzolra!
  - Pl.: i. vevő egy <szín> színű, <méret> méretű pónit szeretne venni. A kért póniból volt raktáron, a vevő megvette a pónit.
  - j. vevő egy <szín> színű, <méret> méretű pónit szeretne venni. A kért póni már sajnos elfogyott ☹

A nap végén az eladók átnézik a raktár tartalmát.

- Írd ki a raktár maradék tartalmát a rendezett sorrendben!
- Ha nem maradt áru, írd ki ezt.

# Házifeladat hf06

Írd meg a `bool rbtree_check(const BST*)` függvényt, amely kap egy bináris kereső fát és ellenőrzi, hogy lehet-e egy érvényes piros-fekete fa. Figyelem, lehet egy sima bináris kereső fa véletlen olyan alakú, hogy létezik színezése, amivel érvényes piros-fekete fa lenne, ezeket is szeretnénk megtalálni.

Ehhez kapcsolódóan a hét folyamán lesz elérhető kiadott kód és teszt esetek.