**Piros-Fekete Keresőfa**

A piros-fekete keresőfa egy önegyensúlyozó bináris keresőfa, mely garantálja, hogy a beszúrás keresés és törlés is O(log n) időben fog lefutni. Egy bináris fa önegyensúlyozó csakis kor lehet, ha bizonyos szabályokat követ, a piros fekete keresőfának 4 fontos “ szabálya”/tulajdonsága van, amit minden művelet után meg kell tartania, csak így garantálhatja a hatékonyságot.

Ez a 4 tulajdonság a következő: a gyökér fekete, piros csomópontoknak fekete leszármazottjaik lehetnek, a gyökértől indulva a fa minden ágán a fekete csomópontok száma megegyezik, a fa levelei feketék és NULL értékük van.

A piros-fekete keresőfa csomópont típusú adattagokból épül fel ,melyek egymáshoz vannak láncolva. A fa típusú adat tartalmaz egy csomópont típusú elemet, mely minden más adathoz el képes vezetni, hisz a csomópont típusú adat felépítése: Természetesen színnel rendelkezik, mely piros vagy fekete, bool változó jelzi, mivel mindkét esetben két lehetőség van. Tartalmaz egy kulcsot is, ami long long típusú, így rengeteg adatot tárolhat, fedések nélkül, ezek a kulcsok segítenek az elem elhelyezésében és keresésében, hisz mivel bináris fáról beszélünk, minden csomópontnak két leszármazottja van, jobb és baloldali, balra a kisebb található, jobbra a nála nagyobb kulcsú adattag. Ezeken kívül tartalmaz egy nil bool típusú mezőt, mely true, akkor és csakis akkor ha levélről van szó, és egy mutatót, mely a szülőre mutat, így a fa bejárása nem csak egy irányú.

Annak érdekében, hogy a fában ne csak az adat kulcsát lehessen tárolni, és hogy skatulyát se húzzunk az adatszerkezetre, a csomópontnak fenntartott egy típus nélküli mező, melyet a felhasználó saját akarata szerint használhat, int ,string,struct, stb. típusú adat tárolására. Ez elősegíti a tárolt adatok mennyiségét, sokféleségét, ugyanakkor mégis megmarad a gyors adatlekérés.

**Csomópont:**

* Kulcs
* Szín
* Bal (Csomópont)
* Jobb (Csomópont)
* Nil
* Szülő
* Infó mező (template)

**Fa:**

* Gyökér (Csomópont)
* Esetleg méret

**Műveletek:**

**Létrehozás:**

**Csomópont :**

Nil létrehozás:

Előfeltétel: Létezzen már egy szülő

Utófeltétel: Létre jön a fekete, 0 kulcsú levél

Csomópont létrehozása, adott kulccsal:

Előfeltétel: Nincs

Utófeltétel: Létre jön az adott kulcsú, piros csomópont

Csomópont létrehozása, adott kulccsal és infó mezővel:

Előfeltétel: Nincs

Utófeltétel: Létre jön az adott kulcsú, piros csomópont, aktív infó mezővel

**Fa:**

Létrehozás

Előfeltétel: Nincs

Utófeltétel: Létre jön a fa, gyökere nullptr

**Szúrás/Törlés:**

Beszúrás:

Előfeltétel: létezzen a fa

Utófeltétel: Az adott kulcsú csomópont bekerül a fába, és egyensúlyozza magát

Törlés:

Előfeltétel: létezzen az adott kulcsú elem

Utófeltétel: Az elem kikerül a fából, a piros fekete fa minden tulajdonsága megmarad

**Adatlekérések:**

Csomópont színe:

Előfeltétel: Létezzen a csomópont

Utófeltétel: Visszatéríti a csomópont színét

Csomópont kulcsa:

Előfeltétel: Létezzen a csomópont

Utófeltétel: Visszatéríti a csomópont kulcsát

Csomópont rangja:

Előfeltétel: Létezzen a csomópont

Utófeltétel: Visszatéríti a csomópont rangját a rendezett sorból

Csomópont előtti csomópont:

Előfeltétel: Létezzen a csomópont, a fában legyen legalább 2 csomópont

Utófeltétel: Visszatéríti az adott csomópontnál eggyel kisebb rangú elemet

Csomópont utáni csomópont:

Előfeltétel: Létezzen a csomópont, a fában legyen legalább 2 csomópont

Utófeltétel: Visszatéríti az adott csomópontnál eggyel nagyobb rangú elemet

Csomópont kiírása:

Előfeltétel: Létezzen a csomópont

Utófeltétel: Megjeleníti a csomópont színét és kulcsát a képernyőn

Fa kiírása

Előfeltétel: Létezzen a fa (ne legyen üres)

Utófeltétel: Megjeleníti a fa összes csomópontjának a színét és kulcsát

**Keresés:**

Két féle képpen lehet:

1. Előfeltétel: Létezzen a fa, es benne a keresett kulcsú elem

Utófeltétel: Visszatéríti az adott kulcsú elemet

2. Előfeltétel: Létezzen a fa

Utófeltétel: Igaz/Hamis-at térít vissza

Csere:

Előfeltétel: Létezzen 2 csomópont

Utófeltétel: Felcserélődnek az adatok

**Minimum/Maximum:**

Minimum keresés:

Előfeltétel: Létezzen a fa és ne legyen üres

Utófeltétel: Visszatéríti a legkisebb kulcsú elemet

Maximum keresés:

Előfeltétel: Létezzen a fa és ne legyen üres

Utófeltétel: Visszatéríti a legnagyobb kulcsú elemet

Testvér:

Előfeltétel: Legyen a megadott csomópont a fában

Utófeltétel: Visszatéríti a testvért, ha van