

Adatszerkezetek

08. Keresőfák II

Vekov Géza

2023. április 19.



Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Ismétlés - Bináris keresőfa

Ismétlés - Bináris keresőfa

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bináris fa

- Olyan fa, melyben minden adatelemnek leg több két rákövetkezője van

Bináris keresőfa

Olyan rendezett bináris fa, melyben:

- Az adatelemek mindegyike rendelkezik egy kulccsal
- Minden adatelemre igaz, hogy:
 - Az adatelem bal oldali részfájában levő elemek kulcsai kisebbek az elem kulcsánál
 - Az adatelem jobb oldali részfájában levő elemek kulcsai nagyobbak az elem kulcsánál

Ismétlés - Bináris keresőfa: adatok

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

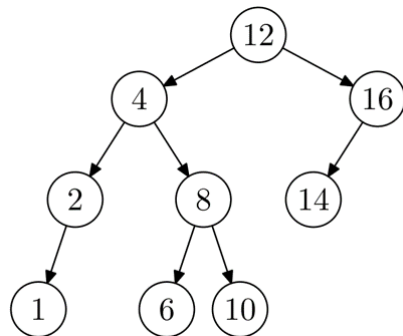
Splay fák

Műveletek

Alkalmazás

Bináris keresőfa elemei

- **Kulcsok** (által jellemzett objektumok)
- Feltételezzük, hogy minden kulcs **egyedi**.
- Nem egyedi kulcsok esetén szükséges egy egyértelmű szabály az azonos kulccsal rendelkező elemek kezelésére



Ismétlés - Kiegyensúlyozott bináris fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Műveletek időbonyolultsága

- Bináris fák esetén, a műveletek futási ideje a fa magasságától függ.

Cél

- Minél gyorsabb műveletek.

Megoldás

- kiegyensúlyozott bináris fa

Ismétlés - Kiegyensúlyozott bináris fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Példák

Kiegyensúlyozott fák a következő adatszerkezetek:

- AVL fa
- Piros-fekete fa
- Splay fa

AVL fák: Forgatás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

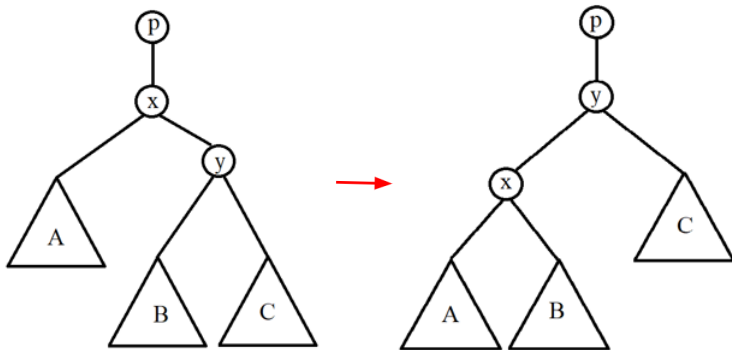
Alkalmazás

Bal forgatás

- Szülő: x
- Jobb gyerek: y

Tudjuk, hogy:

- $x < y$
- $\forall e \in A, e < x < y$
- $\forall e \in B, x < e < y$
- $\forall e \in C, x < y < e$



AVL fák: Forgatás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

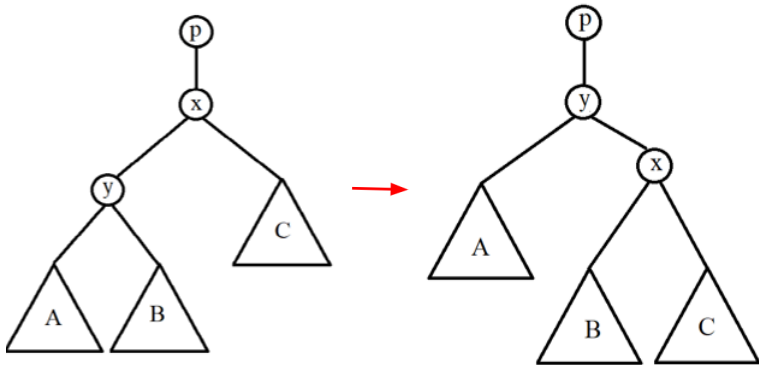
Alkalmazás

Jobb forgatás

- Szülő: x
- Bal gyerek: y

Tudjuk, hogy:

- $x > y$
- $\forall e \in A, e < x < y$
- $\forall e \in B, y < e < x$
- $\forall e \in C, x < y < e$



Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Piros-fekete fák

Definíció

- **A piros-fekete (PF) fák** bináris keresőfák
 - garantálják, hogy a keresés $O(\log n)$ idő alatt hajtható végre

Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Definíció

- **A piros-fekete (PF) fák** bináris keresőfák
 - garantálják, hogy a keresés $O(\log n)$ idő alatt hajtható végre
- A PF fák csomópontjai ki vannak **színezve**: minden csomópont vagy **piros**, vagy **fekete**
- A *NULL* értékű mutatókra úgy tekintünk, mintha a fán kívüli levelekre mutatnának
- A PF fák “*aránylag kiegyensúlyozottnak*” tekinthetők

Megjegyzés

- Rudolf Bayer '72, Leonidas J. Guibas, Robert Sedgewick '78 - véletlenszerűen lett éppen piros-fekete

Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Tulajdonságok

- 0 Minden csomópont piros vagy fekete
- 1 A fa gyökere mindig fekete (CLRS)
- 2 A fa minden (*NULL* értékű) levele fekete
- 3 Piros szülőnek csak fekete gyerekei lehetnek
 - A fában nincs két egymást követő piros csomópont
- 4 Minden csomópont esetén az összes olyan úton, amely az adott csomópontból indul ki és a levélig vezet, ugyanannyi a fekete csomópontok száma.
 - Minden részfa esetén: minden *gyökér-NULL* útvonal ugyanannyi fekete csomópontot tartalmaz

Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

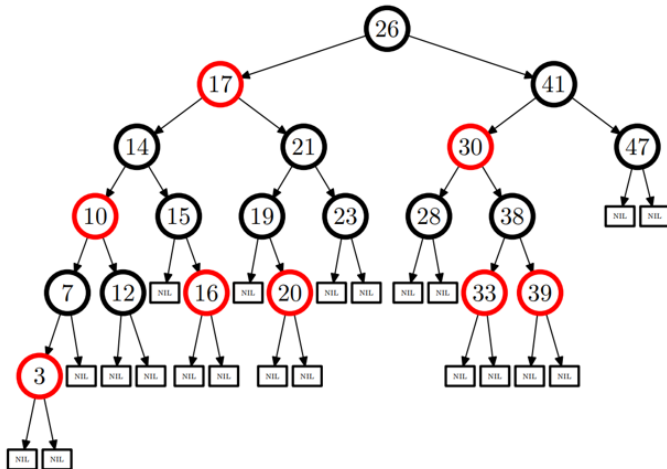
Splay fák

Műveletek

Alkalmazás

Példa

- Gyökér fekete



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

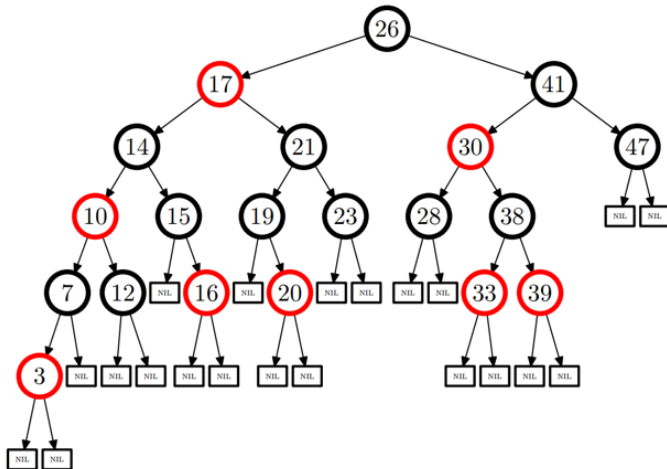
Splay fák

Műveletek

Alkalmazás

Példa

- Gyökér fekete
- Levelek feketék



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

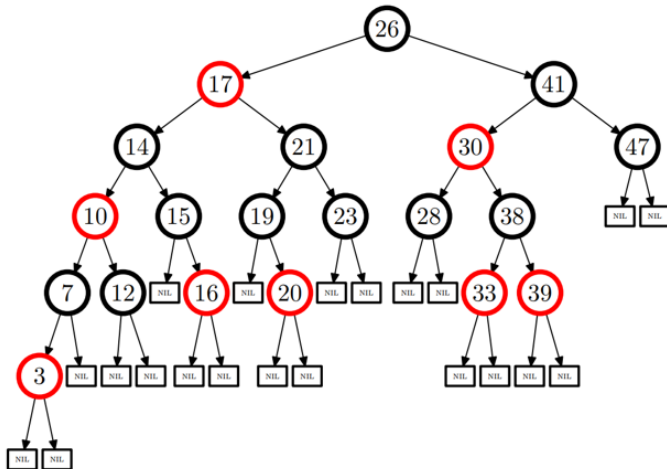
Splay fák

Műveletek

Alkalmazás

Példa

- Gyökér fekete
- Levelek feketék
- Piros csomópontnak nincs piros gyereke



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

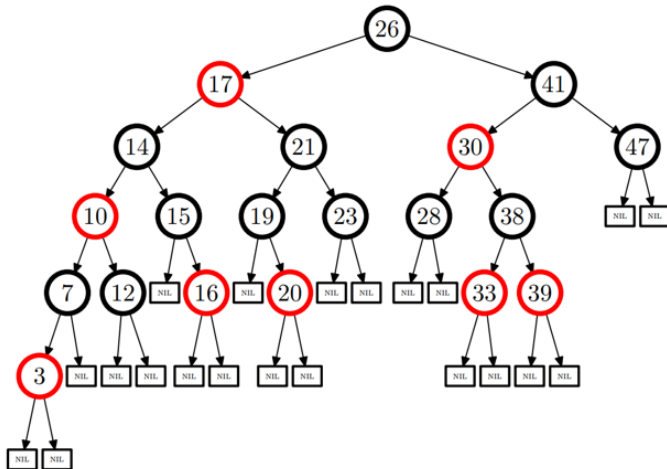
Splay fák

Műveletek

Alkalmazás

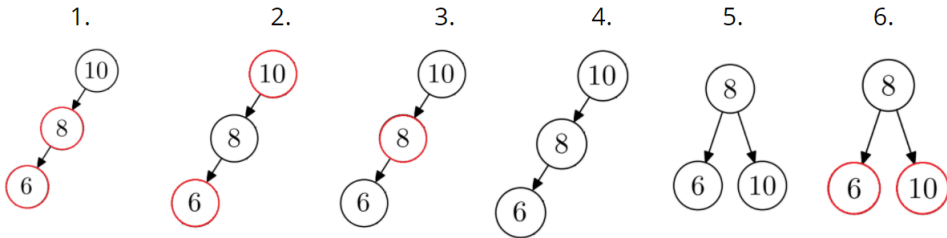
Példa

- Gyökér fekete
- Levelek feketék
- Piros csomópontnak nincs piros gyereke
- Minden gyökér-NULL útvonal ugyanannyi fekete csomópontot tartalmaz



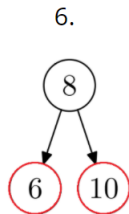
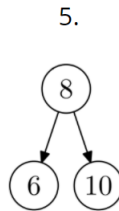
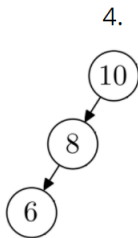
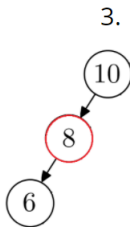
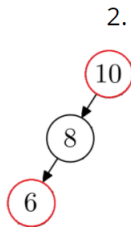
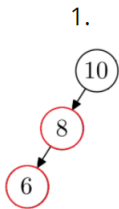
Kérdés

A következő fák közül melyek piros-fekete fák?



Kérdés

A következő fák közül melyek piros-fekete fák?



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

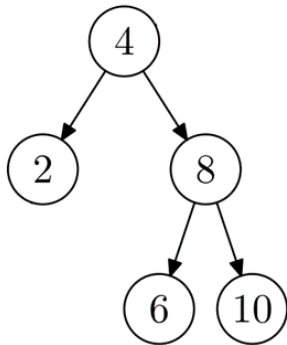
Splay fák

Műveletek

Alkalmazás

Példa

- Hogyan színezzük ki az alábbi bináris fát, hogy PF fát kapjunk?



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

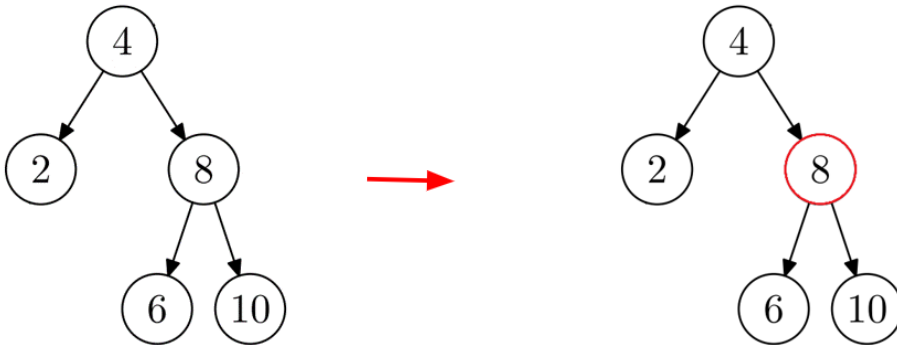
Splay fák

Műveletek

Alkalmazás

Példa

- Hogyan színezzük ki az alábbi bináris fát, hogy PF fát kapjunk?



Piros-fekete fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Definíció

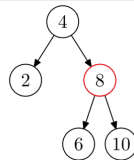
- Egy PF fa x csúcsának **fekete magassága** ($fm(x)$) alatt az x -től ennek egy leszármazott leveléig tartó úton található fekete csúcsok számát értjük, x -et nem beleszámítva.

Példa

Az ábrán látható PF fa **4**-es csúcsának fekete magassága

$$fm(x) = 2,$$

ha figyelembe vesszük, hogy a *NULL* értékek is feketék.



Lemma

- Minden n csomópontot tartalmazó PF fának a **maximális magassága** $2 \log_2(n + 1)$.

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát
 \Rightarrow A csomópontok száma legalább: $2^k - 1$, azaz $n \geq 2^k - 1$

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát
 - ⇒ A csomópontok száma legalább: $2^k - 1$, azaz $n \geq 2^k - 1$
 - ⇒ Mivel minden útvonal legalább k hosszú: $k \leq \log_2(n + 1)$

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát
⇒ A csomópontok száma legalább: $2^k - 1$, azaz $n \geq 2^k - 1$
⇒ Mivel minden útvonal legalább k hosszú: $k \leq \log_2(n + 1)$
- Tehát egy n csomópontot tartalmazó PF fában létezik egy *gyökér-NULL* útvonal, ami legtöbb $\log_2(n + 1)$ fekete csomópontot tartalmaz

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát
⇒ A csomópontok száma legalább: $2^k - 1$, azaz $n \geq 2^k - 1$
⇒ Mivel minden útvonal legalább k hosszú: $k \leq \log_2(n + 1)$
- Tehát egy n csomópontot tartalmazó PF fában létezik egy *gyökér-NULL* útvonal, ami legtöbb $\log_2(n + 1)$ fekete csomópontot tartalmaz
- PF fa, 4. tulajdonság ⇒ minden *gyökér-NULL* útvonal $\log_2(n + 1)$ fekete csomópontot tartalmaz

Piros-fekete fák magassága

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bizonyítás (intuitív módszer)*

- Feltételezzük, egy *gyökér-NULL* útvonal minimum k csomópontot tartalmaz
- Ez azt jelenti, hogy a fa tartalmaz egy $k-1$ magasságú tökéletesen kiegyensúlyozott fát
 \Rightarrow A csomópontok száma legalább: $2^k - 1$, azaz $n \geq 2^k - 1$
 \Rightarrow Mivel minden útvonal legalább k hosszú: $k \leq \log_2(n + 1)$
- Tehát egy n csomópontot tartalmazó PF fában létezik egy *gyökér-NULL* útvonal, ami legtöbb $\log_2(n + 1)$ fekete csomópontot tartalmaz
- PF fa, 4. tulajdonság \Rightarrow minden *gyökér-NULL* útvonal $\log_2(n + 1)$ fekete csomópontot tartalmaz
- PF fa, 3. tulajdonság \Rightarrow minden *gyökér-NULL* útvonal legtöbb $2 \log_2(n + 1)$ csomópontot tartalmaz

* A teljes matematikai bizonyítás matematikai indukció segítségével történik, például Cormen, Leiserson, Rivest, Stein *Új algoritmusok* című kötetében olvasható.

Piros-fekete fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Play fák

Műveletek

Alkalmazás

Piros-fekete fa: Műveletek

- **Keresés(x)** - az x értékű elem keresése
- **Minimum/Maximum** - a PF fa legkisebb/legnagyobb elemének meghatározása
- **Előző/Következő(x)** - az x értékű elem előtti/utáni elem a rendezett sorozatban
- **i . Elem(i)** - a PF fa i . dik legkisebb elemének meghatározása
- **Rang(x)** - az x értékű elem rangjának meghatározása (az x értékű elem "sorszáma" a rendezett sorozatban)
- **Beszúrás(x)** - az x értékű elem beszúrása a PF fába *
- **Törlés(x)** - az x értékű elem törlése a PF fából *
- **Kiírás** - a PF fa rendezett sorrendben való kiírása

* fontos, hogy a keresőfa tulajdonság és a PF tulajdonság megmaradjon

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Kérdés

Szúrjuk be x -et a bináris keresőfába (a már ismert módszerrel).

Kérdés: Melyik PF tulajdonság romolhat el beszúrás során?

- 1 1-es (a gyökér fekete)
- 2 2-es (minden *NULL* levél fekete)
- 3 3-as (piros szülőnek mindig fekete gyerekei vannak)
- 4 4-es (minden gyökér-*NULL* útvonal ugyanannyi fekete csomópontot tartalmaz)
- 5 egyik sem
- 6 mindegyik

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Észrevétel

Beszúrás során sérülhet az **1**-es, a **3**-as vagy a **4**-es PF tulajonság:

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Észrevétel

Beszúrás során sérülhet az **1**-es, a **3**-as vagy a **4**-es PF tulajonság:

- 1** Ha az új csomópontot **feketére** színezzük:
 - lehet egy olyan *gyökér-NULL* útvonalunk, amely eggyel több fekete csomópontot tartalmaz

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Észrevétel

Beszúrás során sérülhet az **1-es**, a **3-as** vagy a **4-es** PF tulajonság:

- 1 Ha az új csomópontot **feketére** színezzük:
 - lehet egy olyan *gyökér-NULL* útvonalunk, amely eggyel több fekete csomópontot tartalmaz
- 2 Ha az új csomópontot **pirosra** színezzük:
 - amennyiben a fa üres volt, a gyökér piros lesz
 - amennyiben az új csomópont szülője piros, akkor két egymásutáni piros csomópontunk lesz

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Észrevétel

Beszúrás során sérülhet az **1-es**, a **3-as** vagy a **4-es** PF tulajonság:

- 1** Ha az új csomópontot feketére színezzük:
 - lehet egy olyan *gyökér-NULL* útvonalunk, amely eggyel több fekete csomópontot tartalmaz
- 2** Ha az új csomópontot pirosra színezzük:
 - amennyiben a fa üres volt, a gyökér piros lesz
 - amennyiben az új csomópont szülője piros, akkor két egymásutáni piros csomópontunk lesz

Döntés

- A 2. eset a kisebbik rossz: sérül az 1. vagy a 3. PF tulajdonság, de ez csak lokális probléma.

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Beszúrás

Beszúrás során sérülhet az **1-es**, a **3-as** vagy a **4-es** PF tulajonság:

- Beszúrjuk az új elemet a PF fába, ugyanúgy mint egy keresőfába
- Az új elemet pirosra színezzük

Több eset fordulhat elő:

1. eset

- A fa továbbra is rendelkezik a piros-fekete tulajdonsággal.
 - nincs teendő

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Beszúrás

Beszúrás során sérülhet az **1-es**, a **3-as** vagy a **4-es** PF tulajonság:

- Beszúrjuk az új elemet a PF fába, ugyanúgy mint egy keresőfába
- Az új elemet pirosra színezzük

Több eset fordulhat elő:

2. eset

- Sérül az 1-es tulajdonság, miszerint a gyökér fekete.
 - Ez abban az esetben fordul elő, ha éppen a gyökeret szúrjuk be (a fa eredetileg üres volt).
 - Átszínezzük a gyökeret feketére.

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Beszúrás

Beszúrás során sérülhet az **1-es**, a **3-as** vagy a **4-es** PF tulajonság:

- Beszúrjuk az új elemet a PF fába, ugyanúgy mint egy keresőfába
- Az új elemet pirosra színezzük

Több eset fordulhat elő:

3. eset

- Sérül a 3-as tulajdonság, miszerint nincs a fában két egymás követő piros csomópont.
 - Ez csak akkor fordulhat elő, ha a beszúrt elem szülője is piros
 - Mivel a gyökér fekete, a beszúrt elemnek biztosan létezik nagyszülője, ami fekete kell legyen (a 3-as tulajdonság miatt)
 - Átszínezéseket és forgatásokat kell végrehajtanunk.

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

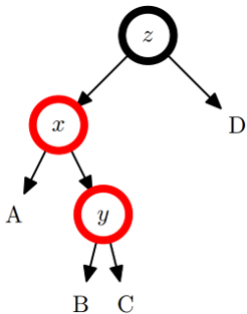
Splay fák

Műveletek

Alkalmazás

3.1 eset - sérül a 3-as tulajdonság

- Az y/x beszúrt elem "nagybácsija" (szülő testvére) fekete
- (Az A, B, C és D részfák fekete gyökerűek - lehet üres fa is)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

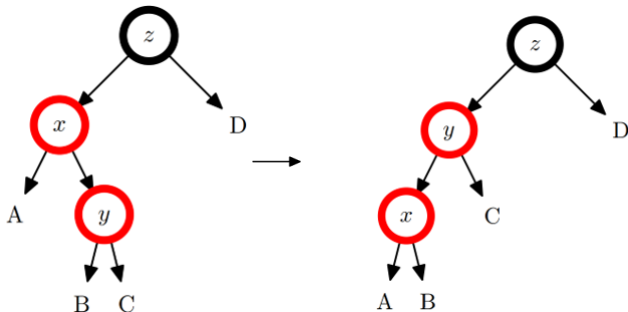
Splay fák

Műveletek

Alkalmazás

3.1 eset - sérül a 3-as tulajdonság

- Az y/x beszúrt elem "nagybácsija" (szülő testvére) fekete
- (Az A, B, C és D részfák fekete gyökerűek - lehet üres fa is)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

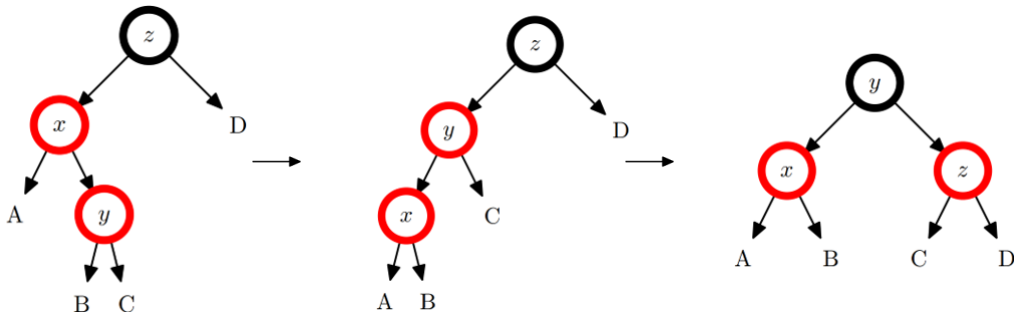
Splay fák

Műveletek

Alkalmazás

3.1 eset - sérül a 3-as tulajdonság

- Az y/x beszúrt elem "nagybácsija" (szülő testvére) fekete
- (Az A, B, C és D részfák fekete gyökerűek - lehet üres fa is)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

3.1 eset - sérül a 3-as tulajdonság

- Az y/x beszúrt elem "nagybácsija" (szülő testvére) fekete
 - (Az A, B, C és D részfák fekete gyökerűek - lehet üres fa is)
- Legtöbb két forgatással helyreállítjuk a PF tulajdonságot

Megjegyzés

- Hasonlóan járunk el, ha a szülő *jobb* leszármazott.
- A középső ábra külön esetként is létezhet.

Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

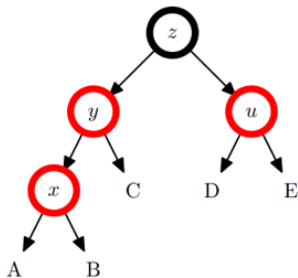
Splay fák

Műveletek

Alkalmazás

3.2 eset - sérül a 3-as tulajdonság

- Az x beszúrt elem "nagybácsija" (szülő testvére) piros
- (Az A, B, C, D és E részfák fekete gyökerűek - lehet üres fa is)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

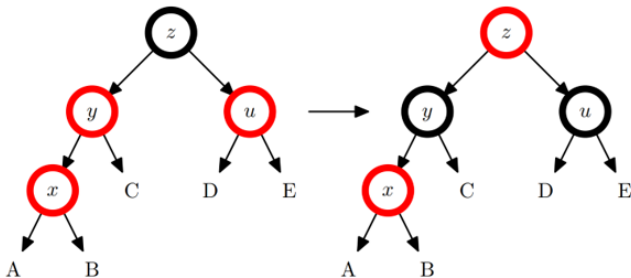
Splay fák

Műveletek

Alkalmazás

3.2 eset - sérül a 3-as tulajdonság

- Az x beszúrt elem "nagybácsija" (szülő testvére) piros
- (Az A, B, C, D és E részfák fekete gyökerűek - lehet üres fa is)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

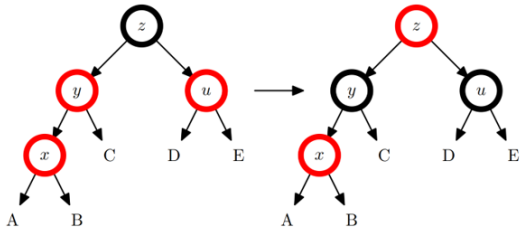
Műveletek

Alkalmazás

3.2 eset - sérül a 3-as tulajdonság

- Az x beszúrt elem "nagybácsija" (szülő testvére) piros
- (Az A, B, C, D és E részfák fekete gyökerűek - lehet üres fa is)

- Elég átszínzeni a csomópontokat
- Nincs szükség forgatásokra



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

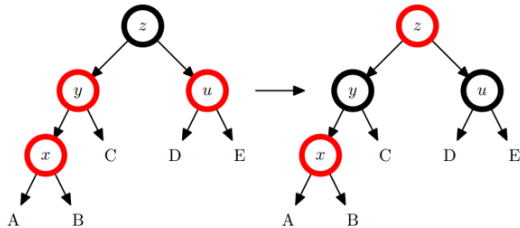
Splay fák

Műveletek

Alkalmazás

3.2 eset - az átszínezés után

- nem változik a levelekbe vezető utakon a fekete elemek száma (4-es tulajdonság)



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

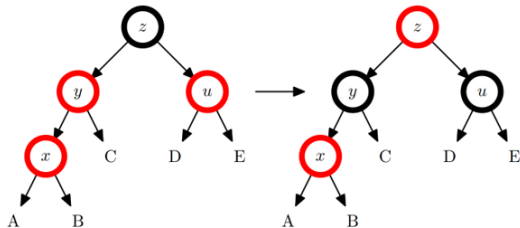
Splay fák

Műveletek

Alkalmazás

3.2 eset - az átszínezés után

- nem változik a levelekbe vezető utakon a fekete elemek száma (4-es tulajdonság)
- 3-as tulajdonság viszont továbbra is sérülhet, ha z szülője piros \Rightarrow feljebb tolhatjuk a két egymást követő piros csomópontot



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

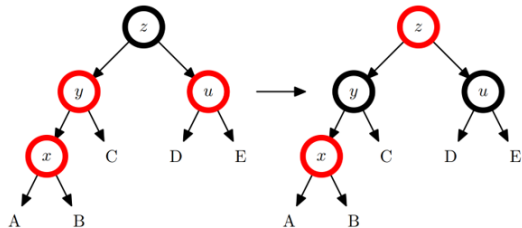
Alkalmazás

3.2 eset - az átszínezés után

- nem változik a levelekbe vezető utakon a fekete elemek száma (4-es tulajdonság)
- 3-as tulajdonság viszont továbbra is sérülhet, ha z szülője piros \Rightarrow feljebb toltuk a két egymást követő piros csomópontot

Az eljárást addig ismételjük, amíg

- z szülője fekete nem lesz
- z gyökér nem lesz
 - átszínezzük a gyökeret feketére
- z nagybácsija fekete nem lesz (3.1 eset)
 - egy vagy két forgatással helyreállítjuk a tulajdonságot



Piros-fekete fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Bonyolultság - PF tulajdonság visszaállítása

- 1. eset: nem romlik el a PF tulajdonság $\rightarrow O(1)$
- 2. eset: üres fába történő beszúrás $\rightarrow O(1)$
- 3.1. eset: $\rightarrow O(1)$
 - sérül a 3-as tulajdonság, fekete "nagybácsi"
 - max. 2 forgatás
- 3.2. eset: $\rightarrow O(\log n)$
 - sérül a 3-as tulajdonság, piros "nagybácsi"
 - legrosszabb eset: "feltoljuk" a két egymás utáni piros elemet a gyökérig

Bonyolultság - beszúrás

beszúrás a keresőfába + PF tulajdonság visszaállítása $\rightarrow O(\log n)$

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Törlés

- 1 Megkeressük a nála kisebb legnagyobb vagy nála nagyobb legkisebb elemet, legyen ez y .
- 2 y adattagjait rámásoljuk a törlendő elem adattagjaira.
- 3 y mindig levélelem vagy csak egy leszármazottja van.

Ezután két fő eset fordulhat elő:

Legyen x : y leszármazottja (valós vagy *NULL*). Ekkor:

- **y piros:** kitörölhetjük, nem sérül egyik PF tulajdonság sem.■
- **y fekete:**
 - Ha y gyökér volt, akkor sérül az 1-es tulajdonság.
 - Ha x és *szülő*[y] is piros, akkor sérül a 3-as tulajdonság.
 - Minden olyan út, ami az y -on keresztül haladt egy feketével rövidebb lesz.

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Törlés - y fekete

- Kitöröljük y -t és x , a leszármazottja "megörökli" a színét:
 - Ha x fekete volt, akkor kétszeresen fekete lesz, hogy a 4-es tulajdonság ne sérüljön. Sérül az 0-ás tulajdonság.
 - Ha x piros volt, akkor piros-és-fekete lesz, hogy a 4-es tulajdonság ne sérüljön. Sérül az 0-ás tulajdonság.
- A továbbiakban, ha nincs megjelölve, akkor az x kétszeresen fekete lesz.
- x tulajdonképpen színe nem változik.
- Legyen w : x testvére.

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Feladat

El kell tüntetni az x -hez rendelt plusz feketét:

- Az aktuális x piros-fekete csúcsra mutat, és feketére színezzhetjük.
- Az aktuális x a gyökerre mutat, így elhagyhatjuk a plusz feketét.
- Forgatásokkal és színezésekkel alakítjuk át a fát úgy, hogy az előző két eset valamelyikét kapjuk.

Megjegyzés

- A műveletek szimmetrikusai is léteznek.
- A c és c' színek tetszőlegesen jelölhetnek pirosat vagy feketét, az algoritmus szempontjából nem relevánsak.

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

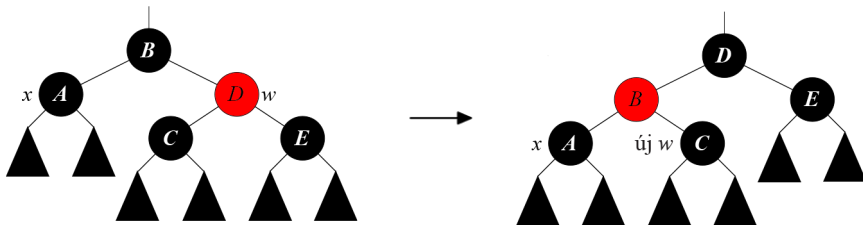
Splay fák

Műveletek

Alkalmazás

1. eset

- w színe piros



- Mivel az új w az eredetinek egy gyereke lesz, a transzformálást követően a 2., 3. vagy 4. esethez jutunk, ahol w fekete.
- x még mindig kétszeresen fekete.

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

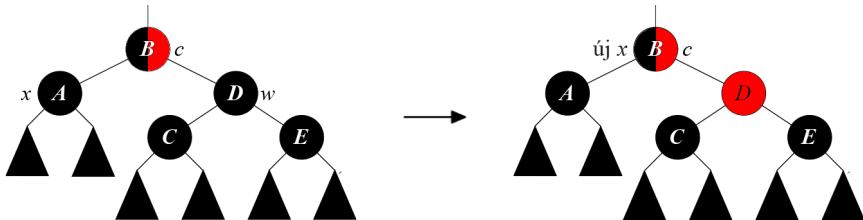
Splay fák

Műveletek

Alkalmazás

2. eset

- w színe fekete
- w mindkét gyereke fekete



- x -től és w -től is "elveszünk" egy feketét, és B -nek $szülő[x]$ -nek adjuk.
- Ha az 1. eset után hajtjuk végre, akkor $szülő[x]$ piros volt, feketére színezzük és lejárt az algoritmus.■

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

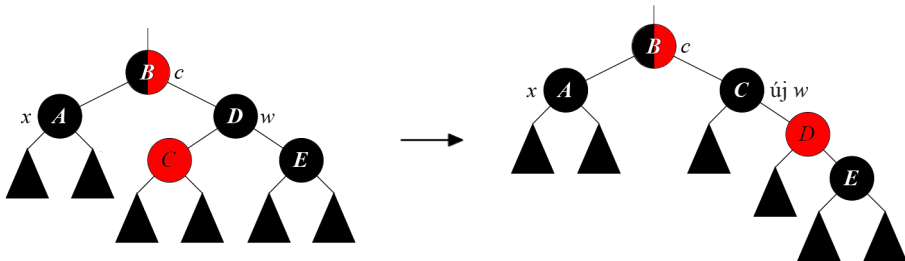
Splay fák

Műveletek

Alkalmazás

3. eset

- w színe fekete
- w jobb gyereke fekete, bal gyereke piros



- Felcseréljük a C és D csúcsok színét, majd jobbra forgatunk.

- 4. esetre transzformál.

Piros-fekete fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

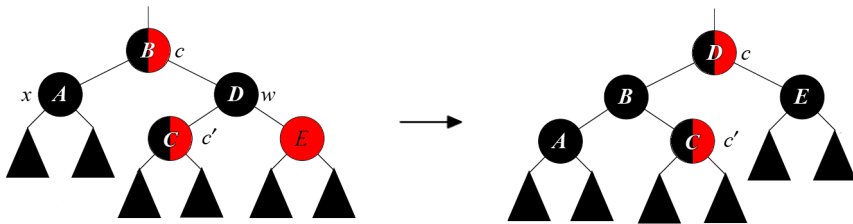
Splay fák

Műveletek

Alkalmazás

4. eset

- w színe fekete
- w jobb gyereke piros



- megváltoztatjuk a w , $szülő[x]$ és $jobb[w]$ csúcsok színét, majd balra forgatunk B körül.
- x -ről levettük a kétszeres feketét, és a következő x érték a gyökér.■

Completely Fair Scheduler

- *Linux kernel alapértelmezett ütemezési algoritmus*
- Minden várakozó folyamat egy PF-fába kerül
- A PF fa kulcsa: a folyamat végrehajtási ideje
- Minden folyamat esetén meghatározunk egy maximum futási időt is
- Ütemezés:
 - Minden lépésben kiválasztjuk a PF fa minimumát (legrövidebb idő alatt ér véget).
 - Ha a folyamat sikeresen lefut:
 - töröljük a PF fából
 - Ha a folyamat eléri a maximális futási idejét / vagy külső hatásra felfüggesztődik:
 - a folyamatot leállítjuk
 - töröljük és újra beszúrjuk a PF fába a folyamatot az új végrehajtási idejével
 - Addig ismételjük, amíg ki nem ürül a PF fa

Piros-fekete fa: alkalmazás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Java 8 - **HashMap** / C++ STL - **map**

- Hash tábla Javaban / C++-ban
- Ütközések kezelésére *láncolást* használ: több azonos hash-értékű elem lehet egy helyen
- Láncolt lista helyett: PF fa
 - Keresési idő: $O(n)$ helyett $O(\log n)$

Piros-fekete fa - AVL fa

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Elmélet

- Ugyanazok a lehetséges műveletek
- Ugyanaz a futási idő: $O(\log n)$

Gyakorlat

- A piros-fekete fák csak "aránylag kiegyensúlyozottak"
 - a keresés enyhén gyorsabb az AVL fában
- Az AVL fádba való beszúrás több forgatással jár
 - A beszúrás enyhén gyorsabb a PF fába

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Splay fák

Splay fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Alapötlet

- **Önszervező** bináris keresőfa
- Ha a felhasználói igények teljesítése közben eljutunk valahova, akkor nem távozunk onnan "csak úgy", hanem rászánunk még valamennyi időt, hogy alakítsuk a környéket ahová kerültünk.

Cél

- A gyakran keresett elemek "kéznél" legyenek

Megjegyzés

- Daniel Sleator, Robert Tarjan '83

Splay fák

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Tulajdonságok

- Egy átlagos alkalmazásban a keresések 80%-a az elemek 20%-ra vonatkozik (*Pareto elv*)

Tulajdonságok

- Egy átlagos alkalmazásban a keresések 80%-a az elemek 20%-ra vonatkozik (*Pareto elv*)
- A splay fák a keresési műveletek során változtatják az alakjukat:
 - a gyakran keresett elemek a fa "tetején" (a gyökérhez közel) helyezkednek el, így ezekhez gyorsan hozzá lehet férni

Tulajdonságok

- Egy átlagos alkalmazásban a keresések 80%-a az elemek 20%-ra vonatkozik (*Pareto elv*)
- A splay fák a keresési műveletek során változtatják az alakjukat:
 - a gyakran keresett elemek a fa "tetején" (a gyökérhez közel) helyezkednek el, így ezekhez gyorsan hozzá lehet férni
- A splay fák nem feltétlenül kiegyensúlyozottak:
 - Gyakran keresett elemek esetén a keresési idő *konstans*
 - Ritkán keresett elemek esetén a keresési idő *lineáris is lehet*
 - Átlagos keresési idő: $O(\log n)$

Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Splay fa: Műveletek

- **Keresés(x)** - az x értékű elem keresése
- **Minimum/Maximum** - a splay fa legkisebb/legnagyobb elemének meghatározása
- **Előző/Következő(x)** - az x értékű elem előtti/utáni elem a rendezett sorozatban
- **i . Elem(i)** - a splay fa i . dik legkisebb elemének meghatározása
- **Rang(x)** - az x értékű elem rangjának meghatározása (az x értékű elem "sorszama" a rendezett sorozatban)
- **Összeillesztés(x)** - két splay fa összeillesztése *
- **Szétvágás(x)** - az x értékű elem alapján a splay fa szétvágása *
- **Beszúrás(x)** - az x értékű elem beszúrása a splay fába *
- **Törlés(x)** - az x értékű elem törlése a splay fából *
- **Kiírás** - a splay fa rendezett sorrendben való kiírása

Splay fa: keresés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Keresés

- 1 Keresés a bináris keresőfában: megkeressük x -et
- 2 Splay művelet:
 - az x elem fontossága nőtt
 - az x elemet forgatások sorozatával a gyökérbe mozgatjuk

Megjegyzés

- A módszer az x -hez vezető kereső út mentén levő elemeket is közelíti a gyökérhez.
- A gyakran használt elemek közel lesznek a fa tetejéhez.
- A kevésbé "aktív" elemek a levelek felé vándorolnak.

Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Splay(x)

- Forgatások sorozata
 - **Cél:** a keresett csomópont legyen az új gyökér
- Legyen
 - **x:** a keresett csomópont
 - **y:** x szülője
 - **z:** y szülője (x nagyszülője)
- Minden lépés az alábbi feltételektől függ:
 - x bal, vagy jobb gyereke y -nak
 - y gyökér-e vagy sem
 - y bal, vagy jobb gyereke z -nek

Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

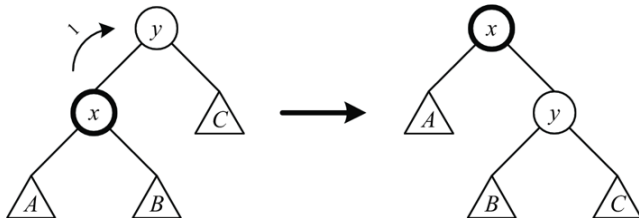
Alkalmazás

Splay(x)

1 **ZIG** lépés (jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y gyökér



Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

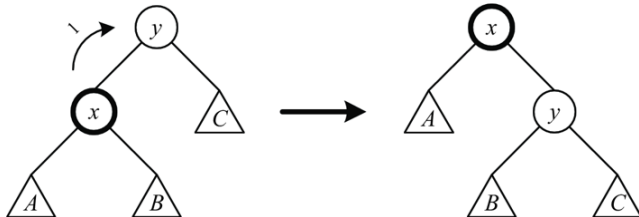
Alkalmazás

Splay(x)

1 **ZIG** lépés (jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y gyökér



Megjegyzés

- Hasonlóan létezik **ZAG** lépés is (bal forgatás)

Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

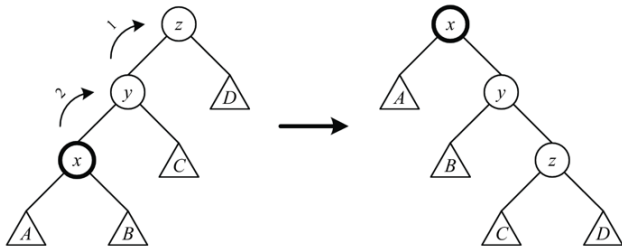
Alkalmazás

Splay(x)

2 ZIGZIG lépés (jobb + jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y bal gyereke z -nek



Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

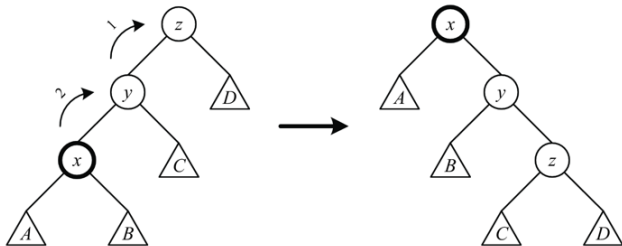
Alkalmazás

Splay(x)

2 ZIGZIG lépés (jobb + jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y bal gyereke z -nek



Megjegyzés

- Hasonlóan létezik **ZAGZAG** lépés is (bal + bal forgatás)

Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

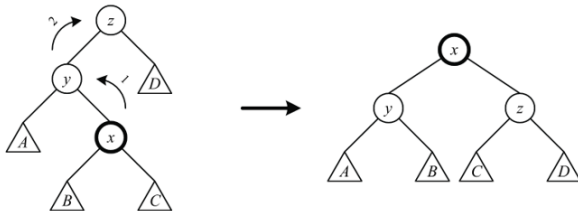
Alkalmazás

Splay(x)

3 **ZIGZAG** lépés (bal + jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y jobb gyereke z -nek



Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

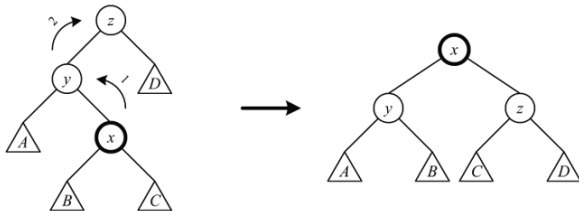
Alkalmazás

Splay(x)

3 ZIGZAG lépés (bal + jobb forgatás)

Feltételek

- x bal gyereke y -nak
- y jobb gyereke z -nek



Megjegyzés

- Hasonlóan létezik **ZAGZIG** lépés is (jobb + bal forgatás)

Splay fa: splay művelet

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Splay(x)

- Addig kell ismételni a megfelelő forgatásokat és a szimmetrikus forgatásokat, amíg a keresett elem a gyökérbe kerül.

Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

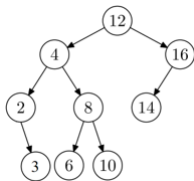
Splay fák

Műveletek

Alkalmazás

Példa

■ Splay(6)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

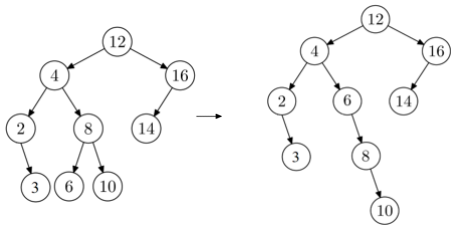
Splay fák

Műveletek

Alkalmazás

Példa

■ Splay(6)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

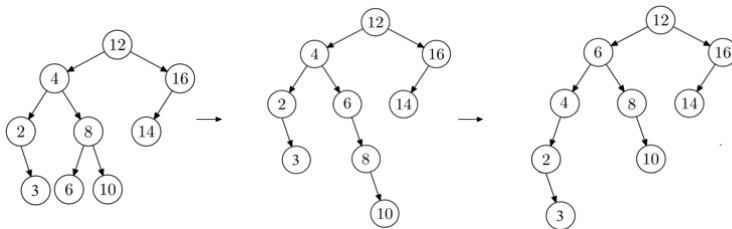
Splay fák

Műveletek

Alkalmazás

Példa

■ Splay(6)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

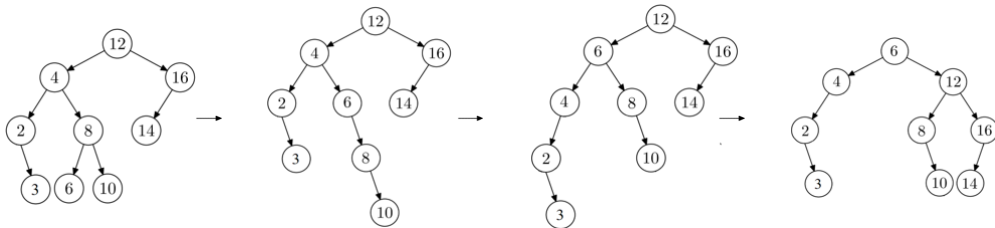
Splay fák

Műveletek

Alkalmazás

Példa

■ Splay(6)



Splay fa: beszúrás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Beszúrás

- 1 Beszúrjuk x -et a bináris keresőfába
- 2 $Splay(x)$ művelet: átrendezzük a fát úgy, hogy x legyen az új gyökér

Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

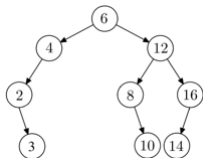
Splay fák

Műveletek

Alkalmazás

Példa

■ Beszúr(5)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

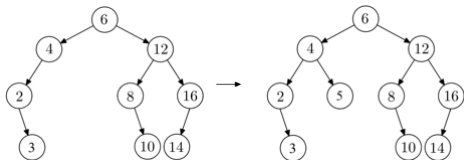
Splay fák

Műveletek

Alkalmazás

Példa

■ Beszúr(5)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

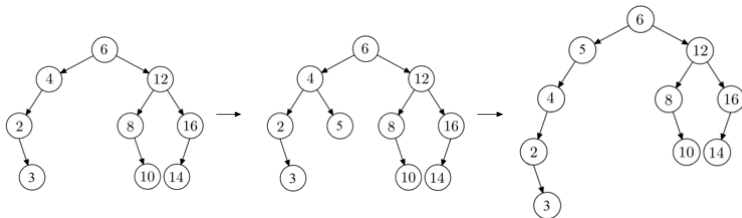
Splay fák

Műveletek

Alkalmazás

Példa

■ Beszúr(5)



Splay fa: műveletek

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

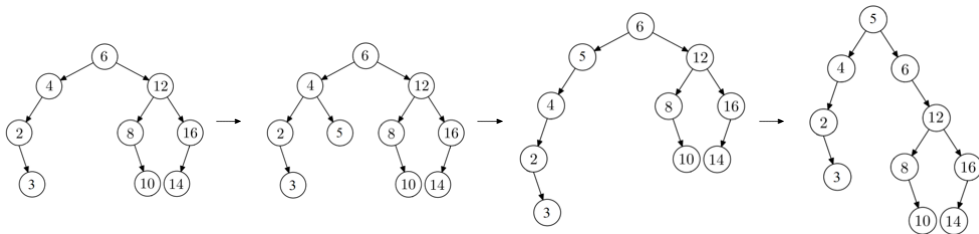
Splay fák

Műveletek

Alkalmazás

Példa

■ Beszúr(5)



Splay fa: összeillesztés és szétvágás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Összeillesztés

A és B splay fát össze lehet illeszteni, ha az egyik (pl. B) valamennyi eleme nagyobb a másik fa valamennyi eleménél.

- 1 Splay(Minimum(B)): az eredmény egy "jobbaldalas" fa, a gyökérnek nincs bal leszármazottja.
- 2 A gyökere a B gyökerének bal leszármazottja lesz.

Szétvágás

A splay fát szét lehet vágni tetszőleges $x \in A$ elem szerint.

- 1 Splay(x): a fa gyökere x lesz.
- 2 A gyökér bal vagy a jobb oldali részfáját le lehet választani.

Splay fa: törlés

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Törlés

- 1 A bináris keresőfáknál ismert törlést alkalmazzuk x -re.
- 2 $\text{Splay}(\text{szülő}(x))$: ha létezik, akkor a törölt elem szülője szerint splay-elünk.

Törlés - változat

- 1 $\text{Splay}(x)$: a fa gyökere x lesz.
- 2 Kitöröljük x -et.
- 3 Összeillesztjük a két megmaradt részét.

Splay fa

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Működés

- az önszerveződésnek köszönhetően a gyökér körül azok a csomópontok csoportosulnak, amelyeket gyakrabban kerestünk

Keresés futási ideje

- Legrosszabb esetben: $O(n)$
- Legjobb esetben: $O(1)$
- Átlagosan: $O(\log n)$

Mikor használják?

- Ha az eltárolt adatoknak egy kisebb részét használjuk rendszeresen.

Splay fa: alkalmazás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Alkalmazás

■ Hálózati routerek:

- Beérkező csomagok továbbítása
- Szükség van egy adatszerkezetre az IP címek eltárolására és gyors keresésre
- Egy IP címre több egymás utáni csomag érkezik hasznos, ha az utoljára használt IP címek könnyen elérhetőek → Splay fa

Splay fa: alkalmazás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Alkalmazás

■ Hálózati routerek:

- Beérkező csomagok továbbítása
- Szükség van egy adatszerkezetre az IP címek eltárolására és gyors keresésre
- Egy IP címre több egymás utáni csomag érkezik hasznos, ha az utoljára használt IP címek könnyen elérhetőek → Splay fa

■ Garbage collection:

- Nem használt memória felszabadítása
- Splay fa: a nem használt objektumok a fa leveleiben lesznek
- Hasonló alkalmazás: cache implementálása

Splay fa: alkalmazás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Alkalmazás

■ Hálózati routerek:

- Beérkező csomagok továbbítása
- Szükség van egy adatszerkezetre az IP címek eltárolására és gyors keresésre
- Egy IP címre több egymás utáni csomag érkezik hasznos, ha az utoljára használt IP címek könnyen elérhetőek → Splay fa

■ Garbage collection:

- Nem használt memória felszabadítása
- Splay fa: a nem használt objektumok a fa leveleiben lesznek
- Hasonló alkalmazás: cache implementálása

■ **sed** - linux parancs

Splay fa: alkalmazás

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Alkalmazás

■ Hálózati routerek:

- Beérkező csomagok továbbítása
- Szükség van egy adatszerkezetre az IP címek eltárolására és gyors keresésre
- Egy IP címre több egymás utáni csomag érkezik hasznos, ha az utoljára használt IP címek könnyen elérhetőek → Splay fa

■ Garbage collection:

- Nem használt memória felszabadítása
- Splay fa: a nem használt objektumok a fa leveleiben lesznek
- Hasonló alkalmazás: cache implementálása

■ **sed** - linux parancs

■ **gcc**

Adatszerkezetek

Vekov Géza

Ismétlés - Bináris
keresőfa

Adatok

Piros-fekete fák

Magasság

Műveletek

Alkalmazás

Splay fák

Műveletek

Alkalmazás

Kérdések

Kérdések? 😊