

Algoritmy a datové struktury

Vícecestné stromy

Obsah přednášky

- ▶ 2-3-4 stromy
- ▶ 2-3 stromy
- ▶ B-stromy

2-3-4 stromy

- ▶ Problémy s častým vyvažováním při vytváření stromu
 - ▶ uvolnění pravidel

2-3-4 stromy

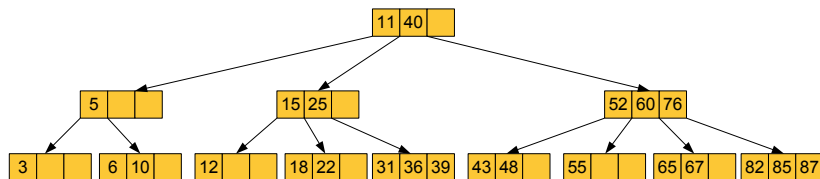
- ▶ Problémy s častým vyvažováním při vytváření stromu
 - ▶ uvolnění pravidel
- ▶ Možnost navázat na uzel více dětí
 - ▶ vícecestné uzly

2-3-4 stromy

- ▶ Problémy s častým vyvažováním při vytváření stromu
 - ▶ uvolnění pravidel
- ▶ Možnost navázat na uzel více dětí
 - ▶ vícecestné uzly
- ▶ 2-3-4 stromy mohou mít
 - ▶ 2,3,4 děti
 - ▶ 1,2,3 datové položky v uzlu

Co se skrývá v názvu?

- ▶ 2,3,4 je počet dětí uzlu
 - ▶ uzel s 1 datovou položkou musí mít 2 děti
 - ▶ uzel s 2 datovými položkami musí mít 3 děti
 - ▶ uzel s 3 datovými položkami musí mít 4 děti
- ▶ Uzel nesmí mít pouze jednoho syna
- ▶ List nesmí být nikdy prázdný
 - ▶ může obsahovat 1,2 nebo 3 datové položky



Další pravidla

- ▶ Pravidla jsou velmi podobná binárnímu vyhledávacímu stromu
- ▶ Označení
 - ▶ datové položky: A, B, C
 - ▶ děti: a, b, c, d

Další pravidla

- ▶ Pravidla jsou velmi podobná binárnímu vyhledávacímu stromu
- ▶ Označení
 - ▶ datové položky: A, B, C
 - ▶ děti: a, b, c, d
- ▶ Všechny klíče v podstromu a musí být menší než klíč A
- ▶ Všechny klíče v podstromu b musí být větší než klíč A a menší než klíč B
- ▶ Všechny klíče v podstromu c musí být větší než klíč B a menší než klíč C
- ▶ Všechny klíče v podstromu d musí být větší než klíč C

Další pravidla

- ▶ Pravidla jsou velmi podobná binárnímu vyhledávacímu stromu
- ▶ Označení
 - ▶ datové položky: A, B, C
 - ▶ děti: a, b, c, d
- ▶ Všechny klíče v podstromu a musí být menší než klíč A
- ▶ Všechny klíče v podstromu b musí být větší než klíč A a menší než klíč B
- ▶ Všechny klíče v podstromu c musí být větší než klíč B a menší než klíč C
- ▶ Všechny klíče v podstromu d musí být větší než klíč C
- ▶ Duplicitní hodnoty nejsou ve stromu povoleny

Zkuste si

- ▶ Napište v libovolném jazyce třídy pro 2-3-4 strom

Zkuste si

- ▶ Napište v libovolném jazyce třídy pro 2-3-4 strom
 - ▶ třída pro strom

```
class Strom234 {  
    Uzel234 koren;  
}
```

Zkuste si

- ▶ Napište v libovolném jazyce třídy pro 2-3-4 strom

- ▶ třída pro strom

```
class Strom234 {  
    Uzel234 koren;  
}
```

- ▶ třída pro uzel stromu

```
class Uzel234 {  
    Polozka[] polozky;  
    int pocet;  
    Uzel234[] deti;  
    Uzel234 rodic;  
}
```

Vyhledávání

- ▶ Velmi podobné jako u binárních stromů
 - ▶ začíná se v kořenu
 - ▶ pokud uzel neobsahuje hledaný klíč, vybere se vhodný potomek a hledání se opakuje
 - ▶ pokud ani v listu hledaný klíč není, pak není ve stromu vůbec
- ▶ Napište v libovolném jazyce metody pro nalezení prvku

Vyhledávání

- ▶ Velmi podobné jako u binárních stromů
 - ▶ začíná se v kořenu
 - ▶ pokud uzel neobsahuje hledaný klíč, vybere se vhodný potomek a hledání se opakuje
 - ▶ pokud ani v listu hledaný klíč není, pak není ve stromu vůbec
- ▶ Napište v libovolném jazyce metody pro nalezení prvku

```
Polozka najdi(Klic klic) {  
    // začneme prohledávat v kořenu  
    Uzel uzel = najdiUzel(koren, klic);  
}
```

Vyhledávání

```
Polozka najdiUzel(Uzel uzel, Klic klic) {
    int i;

    // otestujeme všechny položky v uzlu
    for(i = 0; i<uzel.pocet; i++) {

        // našli jsme klíč
        if(klic == uzel.polozky[i].klic) {
            return uzel.polozky[i];
        }

        // hledaný klíč je menší než klíč položky, nemá cenu hledat dál
        if(klic < uzel.polozky[i].klic) {
            break;
        }
    }

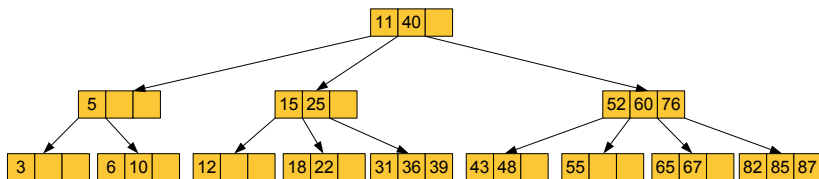
    // klíč v tomto uzlu není, prohledej vhodné dítě (pokud existuje)
    if(uzel.deti[i]!=null) {
        return najdi(uzel.deti[i],klic);
    } else {
        return null;
    }
}
```

Operace vkládání

- ▶ Nová položka se vkládá vždy do listu
 - ▶ až v listu máme jistotu, že podobná hodnota neexistuje
- ▶ Dvě možné situace při hledání místa pro vložení
 - ▶ na cestě není plný uzel
 - ▶ na cestě je plný uzel

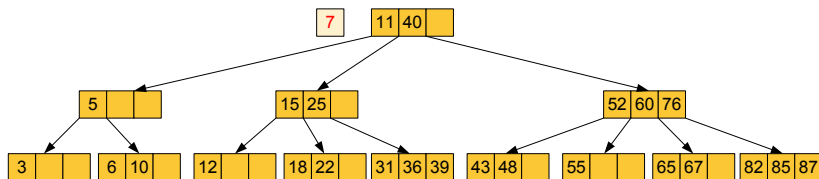
Vkládání – žádný plný uzel

- ▶ Přidání do položky neovlivní strukturu stromu
 - ▶ operace vkládání je jednoduchá
- ▶ Příklad: vložení 7



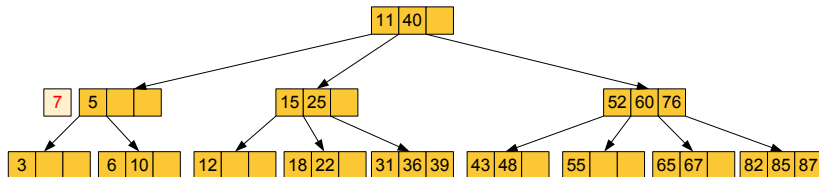
Vkládání – žádný plný uzel

- ▶ Přidání do položky neovlivní strukturu stromu
 - ▶ operace vkládání je jednoduchá
- ▶ Příklad: vložení 7



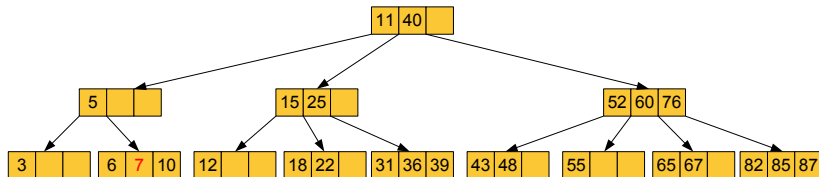
Vkládání – žádný plný uzel

- ▶ Přidání do položky neovlivní strukturu stromu
 - ▶ operace vkládání je jednoduchá
- ▶ Příklad: vložení 7



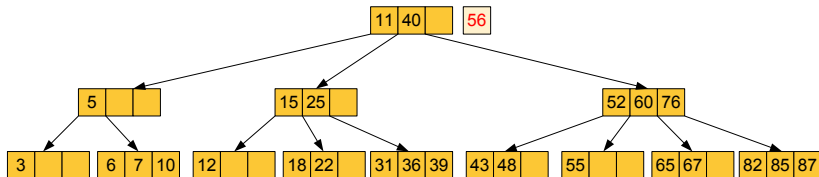
Vkládání – žádný plný uzel

- ▶ Přidání do položky neovlivní strukturu stromu
 - ▶ operace vkládání je jednoduchá
- ▶ Příklad: vložení 7



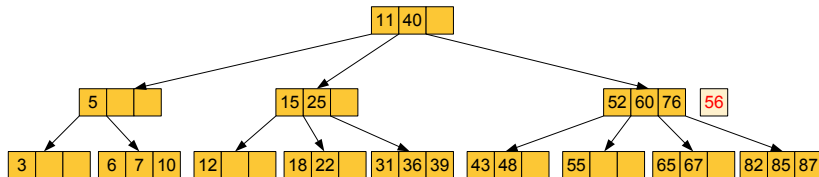
Vkládání – plný uzel

- ▶ Pokud se po cestě najde plný uzel, musí dojít k jeho rozdělení
 - ▶ má za následek vyvážení stromu
- ▶ Jak rozdělit uzel
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. prostřední prvek přesunout do rodičovského uzlu
 3. upravit vazby
- ▶ Příklad: vložení 56



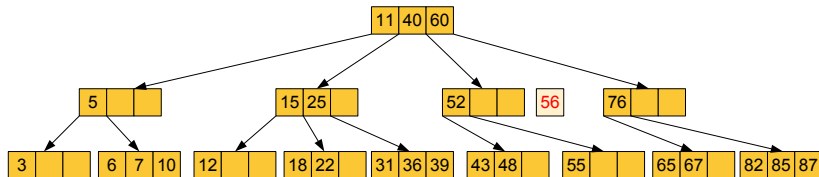
Vkládání – plný uzel

- ▶ Pokud se po cestě najde plný uzel, musí dojít k jeho rozdělení
 - ▶ má za následek vyvážení stromu
- ▶ Jak rozdělit uzel
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. prostřední prvek přesunout do rodičovského uzlu
 3. upravit vazby
- ▶ Příklad: vložení 56



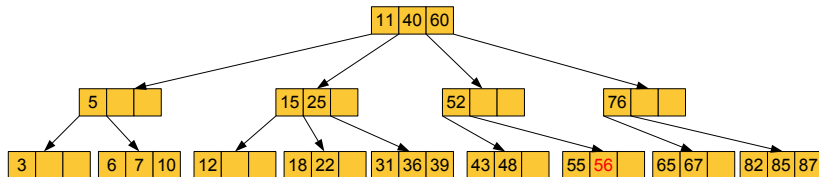
Vkládání – plný uzel

- ▶ Pokud se po cestě najde plný uzel, musí dojít k jeho rozdělení
 - ▶ má za následek vyvážení stromu
- ▶ Jak rozdělit uzel
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. prostřední prvek přesunout do rodičovského uzlu
 3. upravit vazby
- ▶ Příklad: vložení 56



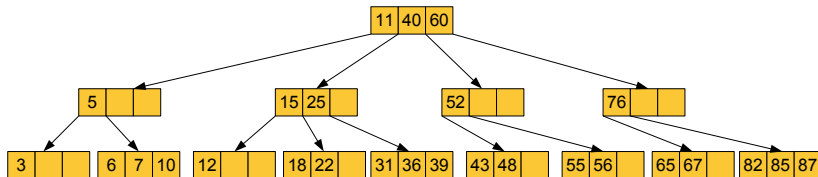
Vkládání – plný uzel

- ▶ Pokud se po cestě najde plný uzel, musí dojít k jeho rozdělení
 - ▶ má za následek vyvážení stromu
- ▶ Jak rozdělit uzel
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. prostřední prvek přesunout do rodičovského uzlu
 3. upravit vazby
- ▶ Příklad: vložení 56



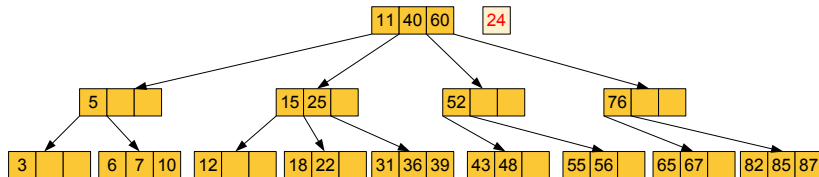
Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



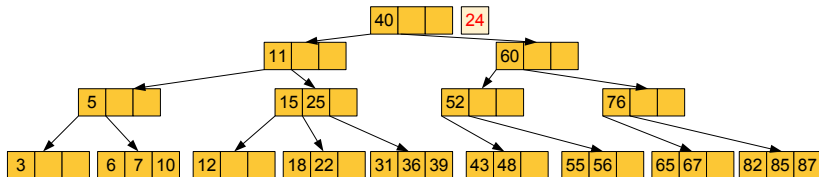
Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



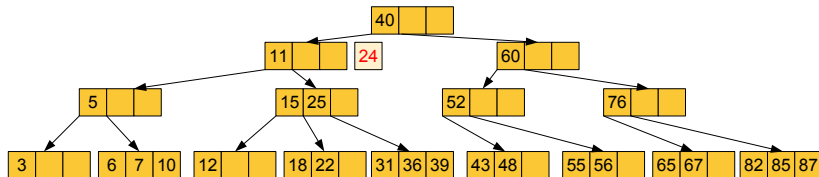
Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



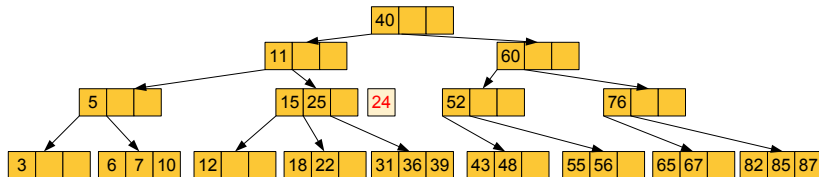
Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



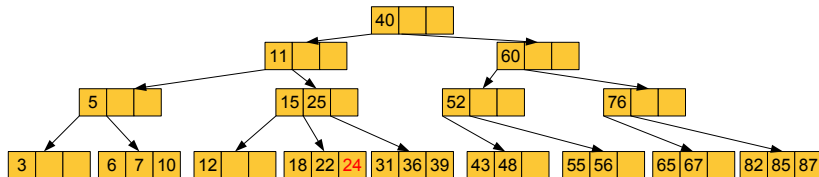
Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



Vkládání – plný kořen

- ▶ Pokud je plný kořen, musí dojít k jeho rozdělení
 - ▶ hloubka stromu se zvětší o 1
- ▶ Jak rozdělit kořen
 1. vytvořit nový uzel a přesunout do něj největší prvek
 2. vytvořit nový kořen a přesunout do něj prostřední prvek
 3. upravit vazby
- ▶ Příklad: vložení 24



Proč rozdělovat cestou dolů?

- ▶ Zjednodušuje práci
 - ▶ rozdělení uzlu (posunutí položky výš) nezpůsobí problém
 - ▶ ve vyšším patře určitě není plný uzel, pokud byl, je rozdělen

Vlastnosti stromu

- ▶ Rychlost
 - ▶ vyvážení Red-Black stromu je $\log_2(N)$
 - ▶ vyvážení ideálního 2-3-4 stromu by bylo $\log_4(N)$, ale
 - ▶ uzly nejsou většinou plné
 - ▶ na test uzlu spotřebujeme více času
 - ▶ rychlosti jsou přibližně stejné
- ▶ Paměť
 - ▶ Red-Black stromy jsou obsazeny skoro na 100%
 - ▶ zjednodušeně 2-3-4 strom má 3+4 reference
 - ▶ v průměru 2 položky na uzel – 2/7 místa se vyhodí
- ▶ Výhoda
 - ▶ mnohem jednodušší kód

2-3 stromy

- ▶ Velmi podobné 2-3-4 stromům
 - ▶ uzly mohou obsahovat 1 nebo 2 položky
 - ▶ uzly ukazují na 2 nebo 3 potomky

2-3 stromy

- ▶ Velmi podobné 2-3-4 stromům
 - ▶ uzly mohou obsahovat 1 nebo 2 položky
 - ▶ uzly ukazují na 2 nebo 3 potomky
- ▶ Operace hledání
 - ▶ probíhá stejně jako u 2-3-4 stromu
 - ▶ změní se pouze počet prohledávaných položek

2-3 stromy

- ▶ Velmi podobné 2-3-4 stromům
 - ▶ uzly mohou obsahovat 1 nebo 2 položky
 - ▶ uzly ukazují na 2 nebo 3 potomky
- ▶ Operace hledání
 - ▶ probíhá stejně jako u 2-3-4 stromu
 - ▶ změní se pouze počet prohledávaných položek
- ▶ Operace vkládání
 - ▶ probíhá stejně jako...

2-3 stromy

- ▶ Velmi podobné 2-3-4 stromům
 - ▶ uzly mohou obsahovat 1 nebo 2 položky
 - ▶ uzly ukazují na 2 nebo 3 potomky
- ▶ Operace hledání
 - ▶ probíhá stejně jako u 2-3-4 stromu
 - ▶ změní se pouze počet prohledávaných položek
- ▶ Operace vkládání
 - ▶ probíhá úplně jinak...

Rozdělení uzlu

- ▶ Co se změnilo?
 - ▶ při rozdělení jsou nutné tři položky
 - ▶ jedena, která zůstane
 - ▶ jedena, která se přesune do sourozence
 - ▶ jedena, která se přesune do rodiče

Rozdělení uzlu

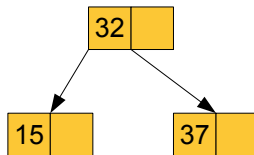
- ▶ Co se změnilo?
 - ▶ při rozdělení jsou nutné tři položky
 - ▶ jedena, která zůstane
 - ▶ jedena, která se přesune do sourozence
 - ▶ jedena, která se přesune do rodiče
 - ▶ jenže
 - ▶ 2-3 strom má v uzlu pouze dvě položky

Rozdělení uzlu

- ▶ Co se změnilo?
 - ▶ při rozdělení jsou nutné tři položky
 - ▶ jedena, která zůstane
 - ▶ jedena, která se přesune do sourozence
 - ▶ jedena, která se přesune do rodiče
 - ▶ jenže
 - ▶ 2-3 strom má v uzlu pouze dvě položky
 - ▶ kde vzít 3 položku?
 - ▶ musí se použít vkládaná položka
 - ▶ nelze upravovat cestou dolů

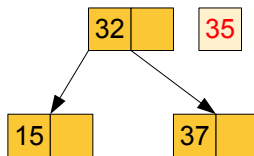
Operace vkládání

- ▶ Vkládání do neplného listu
 - ▶ bez problémů



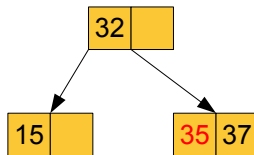
Operace vkládání

- ▶ Vkládání do neplného listu
 - ▶ bez problémů



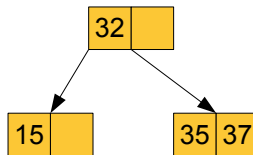
Operace vkládání

- ▶ Vkládání do neplného listu
 - ▶ bez problémů



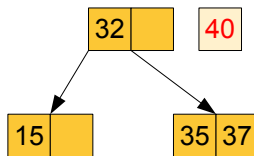
Operace vkládání

- ▶ Vkládání do plného uzlu s neplným rodičem
 - ▶ rozdělení listu, přesunutí prostřední položky do rodiče



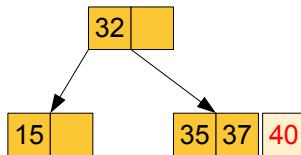
Operace vkládání

- ▶ Vkládání do plného uzlu s neplným rodičem
 - ▶ rozdělení listu, přesunutí prostřední položky do rodiče



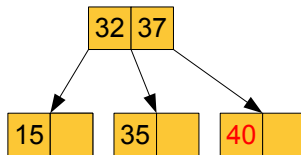
Operace vkládání

- ▶ Vkládání do plného uzlu s neplným rodičem
 - ▶ rozdělení listu, přesunutí prostřední položky do rodiče



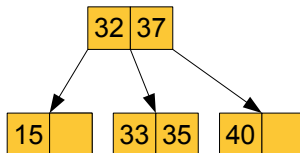
Operace vkládání

- ▶ Vkládání do plného uzlu s neplným rodičem
 - ▶ rozdělení listu, přesunutí prostřední položky do rodiče



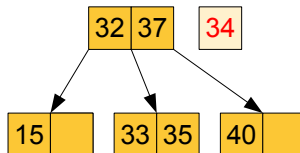
Operace vkládání

- ▶ Vkládání do plného uzlu, s plným rodičem
 - ▶ musí dojít k rozdělení rodiče
 - ▶ případně se musí rozdělit rodič rodiče ...



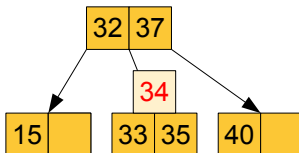
Operace vkládání

- ▶ Vkládání do plného uzlu, s plným rodičem
 - ▶ musí dojít k rozdělení rodiče
 - ▶ případně se musí rozdělit rodič rodiče ...



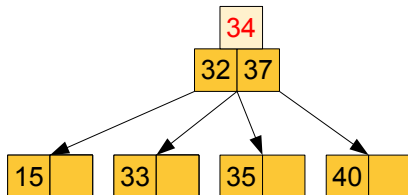
Operace vkládání

- ▶ Vkládání do plného uzlu, s plným rodičem
 - ▶ musí dojít k rozdělení rodiče
 - ▶ případně se musí rozdělit rodič rodiče ...



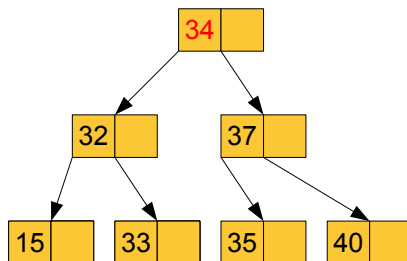
Operace vkládání

- ▶ Vkládání do plného uzlu, s plným rodičem
 - ▶ musí dojít k rozdělení rodiče
 - ▶ případně se musí rozdělit rodič rodiče ...



Operace vkládání

- ▶ Vkládání do plného uzlu, s plným rodičem
 - ▶ musí dojít k rozdělení rodiče
 - ▶ případně se musí rozdělit rodič rodiče ...

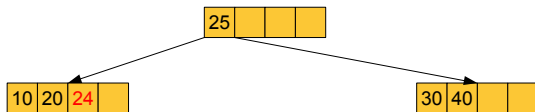


B-stromy

- ▶ Požadované podmínky
 - ▶ každý uzel obsahuje max. n položek
 - ▶ každý uzel (kromě kořene) obsahuje alespoň $n/2$ položek
 - ▶ každý uzel je buď list nebo je počet následovníků $m + 1$, kde m je počet položek
 - ▶ hloubka všech listů je stejná
- ▶ n označuje řád stromu
- ▶ Zajištěno naplnění alespoň na 50%

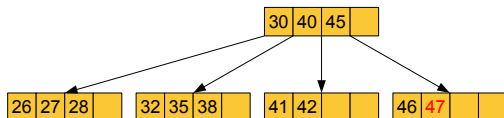
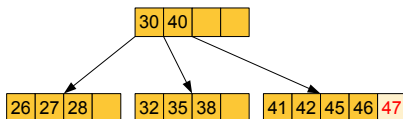
Operace vkládání

- ▶ Podobná jako u 2-3 stromů
 - ▶ položka se vkládá do listu
- ▶ Pokud $m < n$ – vložení bez problémů



Operace vkládání

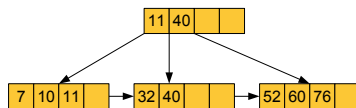
- ▶ Pokud $m = n$
 - ▶ uzel, do kterého položka patří se rozdělí na 2
 - ▶ prostřední položka se přesune o patro výš
 - ▶ v nejhorším případě se rekurzivně pokračuje až ke kořenu



Modifikace a použití

► B+ strom

- všechny hodnoty jsou v listech, vnitřní uzly obsahují pouze klíče
- listy jsou navzájem propojené



► Použití

- souborové systémy (NTFS, XFS)
 - řád stromu většinou odpovídá velikosti diskového bloku
- velké databáze

Konec