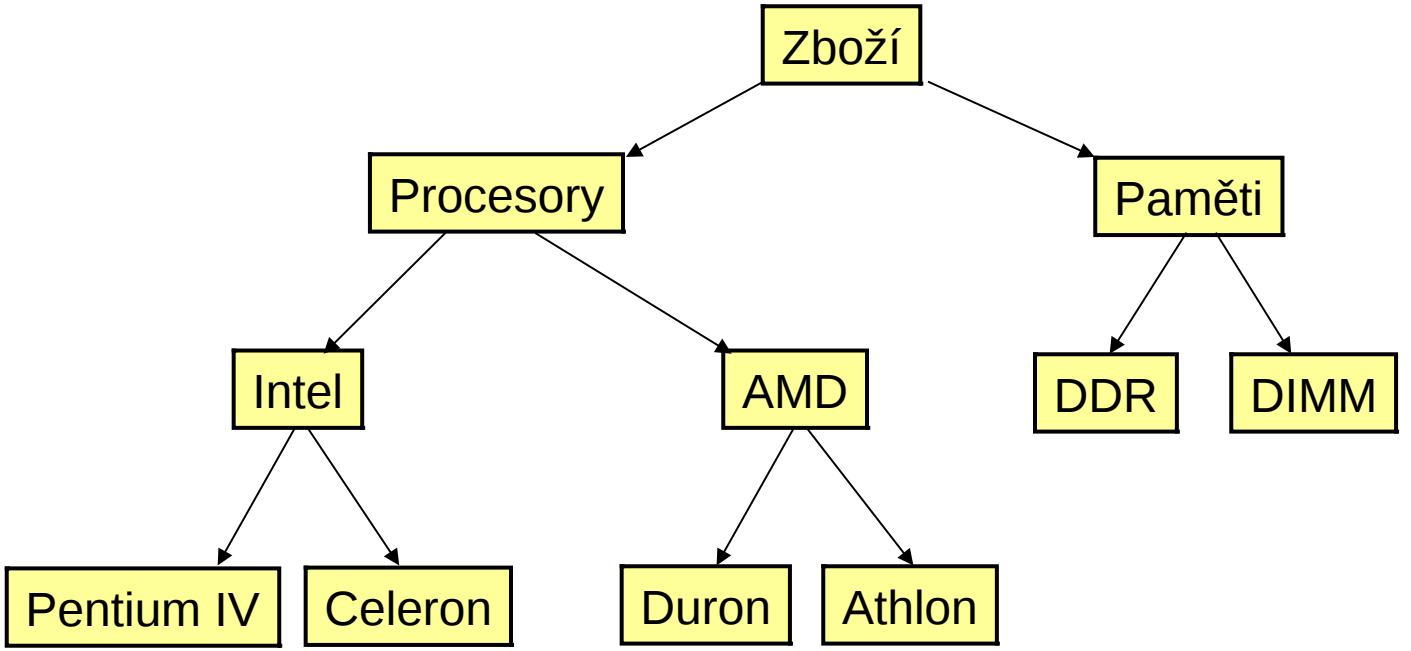
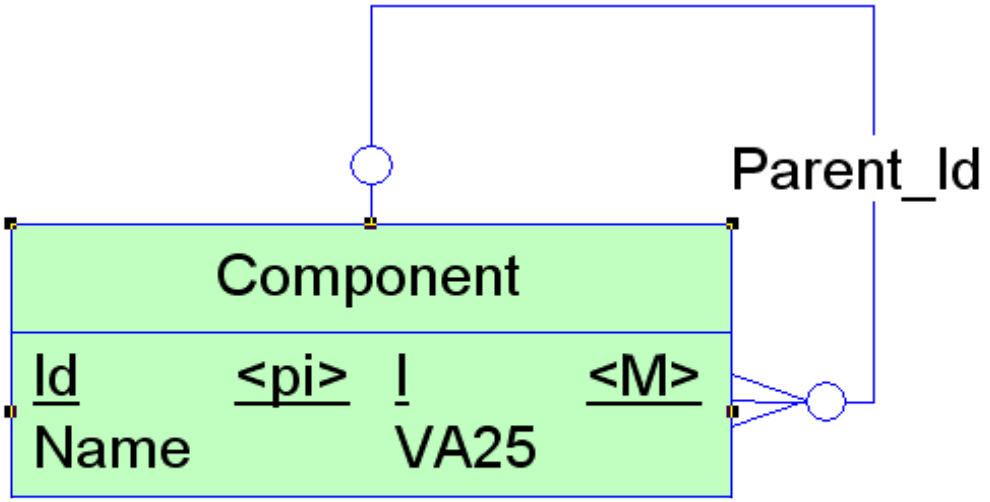
Stromové struktury v relační databázi

Stromové struktury a relační databáze



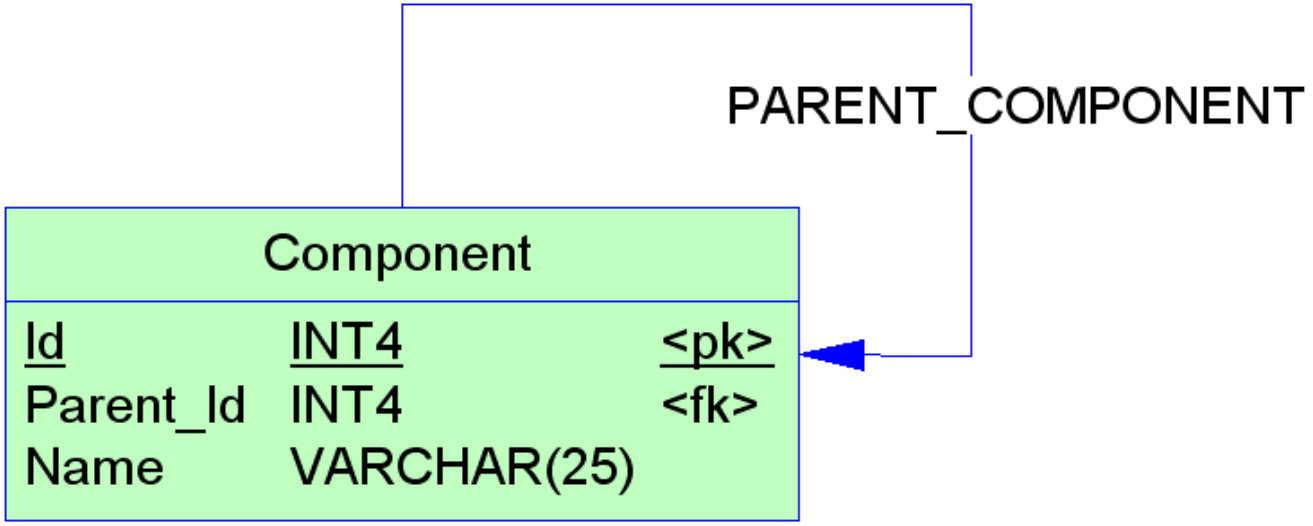
http://interval.cz/clanky/metody-ukladani-stromovych-dat-v-relacnich-databazich/

Stromové struktury a relační databáze



Konceptuální model

Stromové struktury a relační databáze



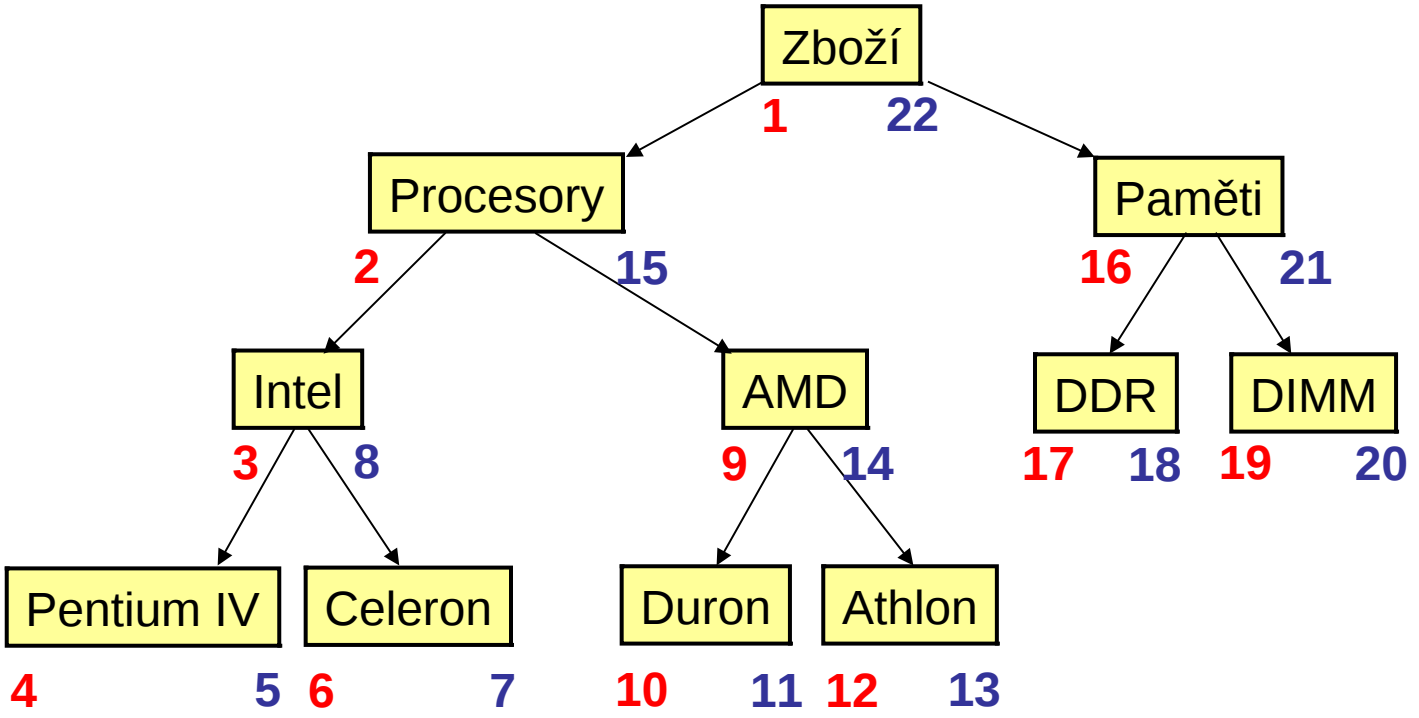
Logický model

Stromové struktury a relační databáze

Hledání všech uzlů z podstromu daného uzlu - rekurze

```
void getTree(int parent) {
  ResultSet rsData = statement.executeQuery(
                "SELECT * FROM TREE WHERE Parent_Id=?1")
           .setParameter(parent);
  while (rsData.next()) {
     System.out.println();
     System.out.println(rsData.getString("Name"));
     parent = rsData.getString("Parent_Id");
     getTree(parent);
  }
}
```

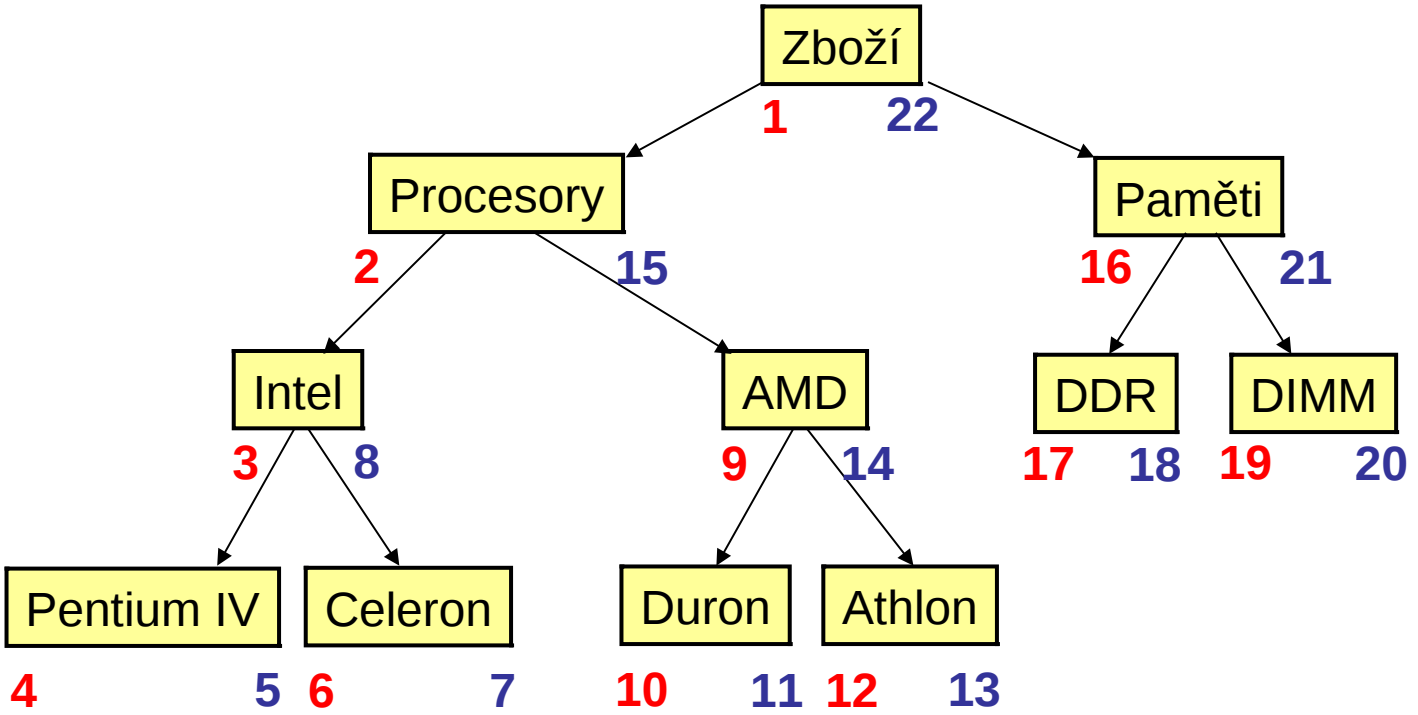
Stromové struktury a relační databáze



Stromové struktury a relační databáze

COMPONENTS				
ID	**NAME**	**PARENT_ID**	**LEFT**	**RIGHT**
1	Kategorie zboží	0	1	22
2	Procesory	1	2	15
3	Intel	2	3	8
4	Pentium IV	3	4	5
5	Celeron	3	6	7
6	AMD	2	9	14

Stromové struktury a relační databáze



SELECT *
FROM *COMPONETS C1, COMPONENTS C2*
WHERE *C1.NAME = "INTEL"* AND
C2.LEFT > C1.LEFT AND
C2.RIGHT < C1.RIGHT

Indexace pomocí B-stromů

Indexace jako prostředek optimalizace výkonu

SELECT *
FROM *PERSON*
WHERE *(GENDER=FEMALE)* AND *(AGE < 32)*

Dotaz tohoto typu bude vykonán mnohem rychleji, bude-li k dispozici index, jehož indexačním výrazem bude {*GENDER*, *AGE*}.

Jednou z hodnot tohoto indexačního výrazu může být například dvojice *<FEMALE, 27>.*

Teorie idexačních technik (vyhledávacích datových struktur) používá pojem **klíč** namísto pojmu **indexační výraz**. Nezaměňovat s **klíčem tabulky** !

CREATE INDEX *PERSON_GENDER_AGE* **ON** *PERSON* (*GENDER*, *AGE*)

Indexace jako prostředek optimalizace výkonu

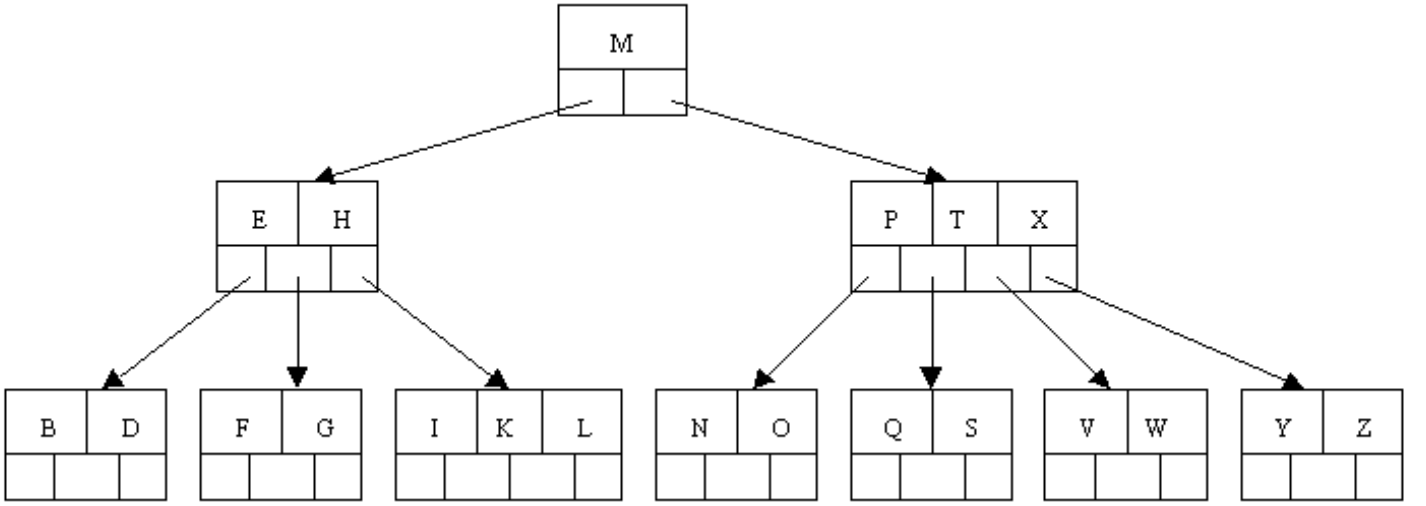
Opatrně s disjunkcemi v podmínkách.

SELECT *
    FROM *PERSON*
    WHERE *(GENDER=FEMALE)* **OR** *(AGE < 32)*

DBMS by měl využít dvou klíčů – a to {*GENDER*} a {*AGE*}

Některé DBMS (např. PostgreSQL) nemusejí pro takovéto dotazy využívat efektivně existující indexy.

Princip B-stromu



B-strom má definovánu:

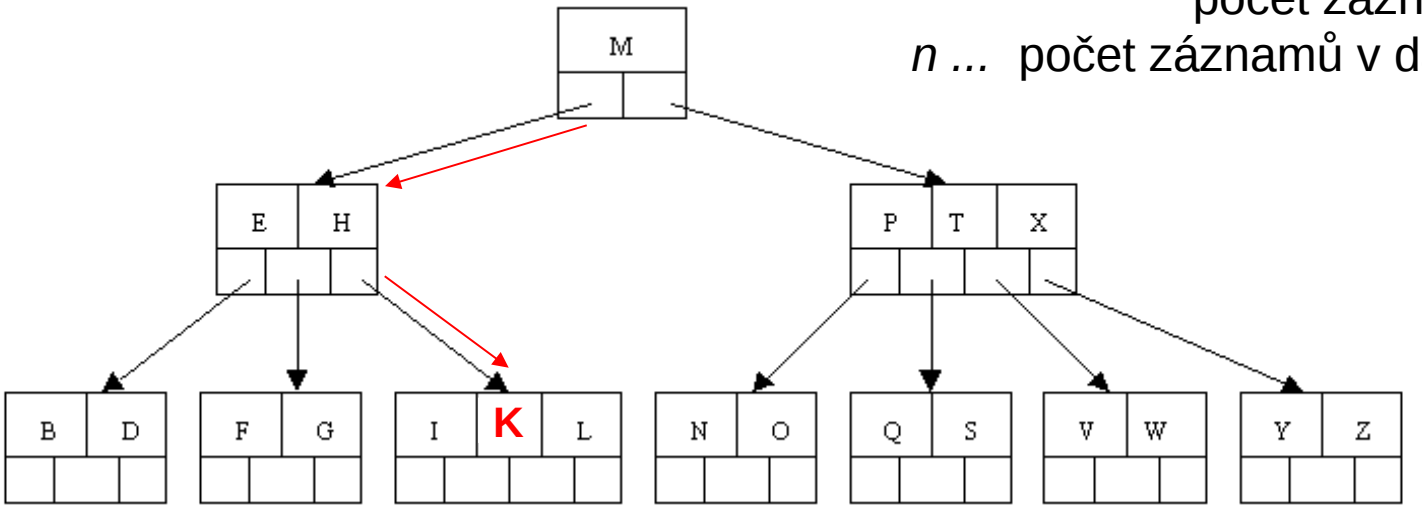
- maximální kapacitu uzlu (max. počet záznamů v uzlu)
- minimální kapacitu uzlu (min. počet záznamů v uzlu)

Záznamy uvnitř uzlu jsou setříděné podle hodnoty klíče.

Princip B-stromu

max(min) … maximální (minimální) počet záznamů v uzlu

n ... počet záznamů v databázi

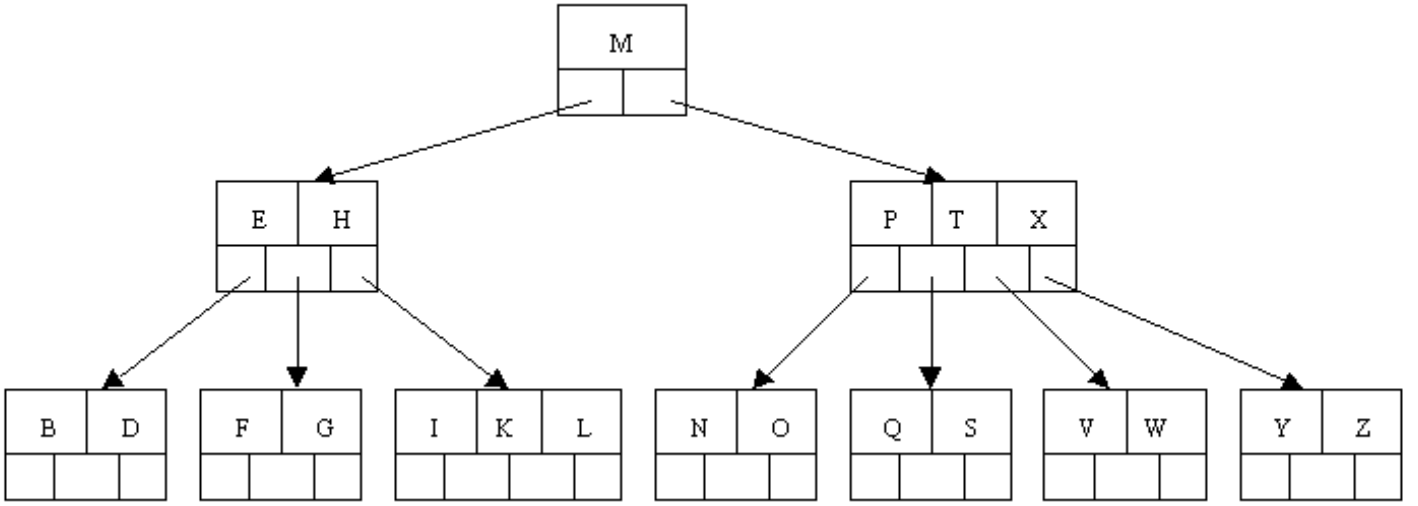


- Každý uzel – stránka v databázi
- Smysl – minimalizace počtu přístupů do databáze
- Hloubka B-stromu

v nejlepším případě (všechny uzly naplněny na 100%) ... $\log_{max} n$

v nejhorším případě (všechny uzly naplněny na *min*) ... $\log_{min} n$

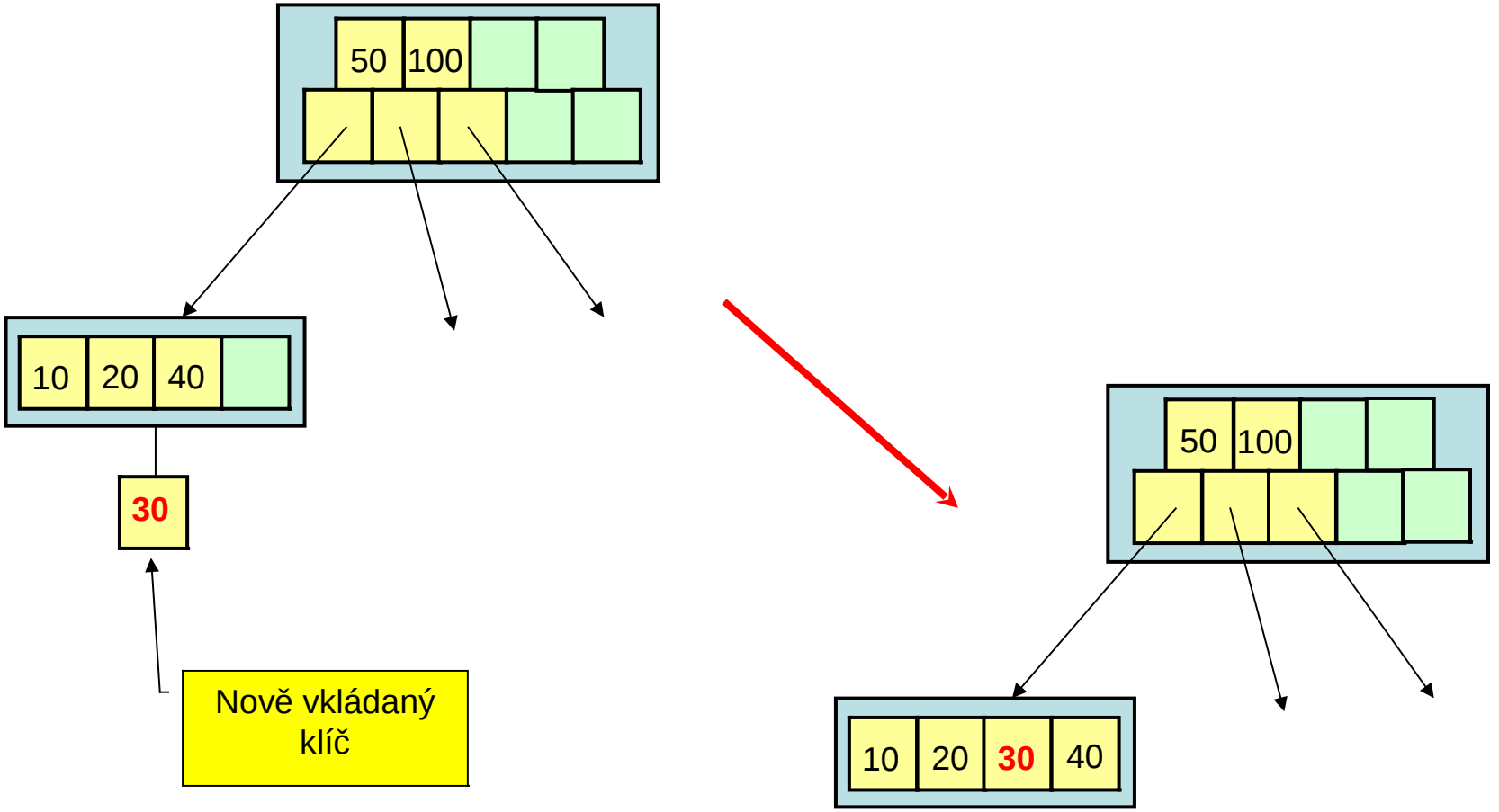
Vkládání do B-stromu



- Každý uzel – stránka v databázi
- Při prvotní konstrukci stromu se uzly naplňují pouze částečně, 25% - 30% kapacity uzly se nechávají volné jako rezerva pro nově vkládané uzly
- Je-li uzel zcela zaplněn a je třeba do něj přidat další záznam, uzel se rozštěpí na 2 zaplněné z 50%. V takovém případě se musí přidat záznam i do příslušného uzlu o patro výš

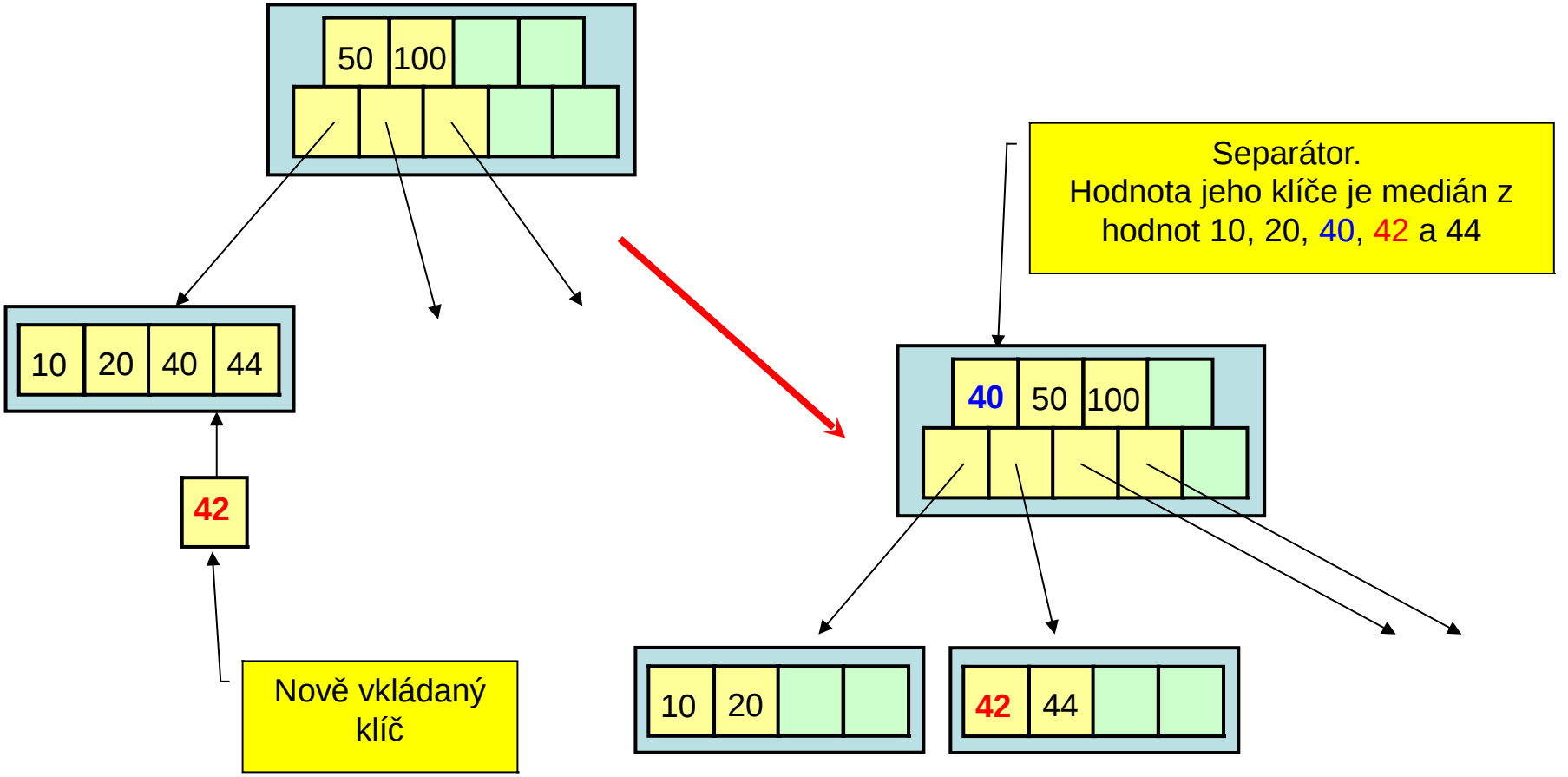
Vkládání záznamu do B-stromu

Triviální, není-li kapacita daného uzlu naplněna



Vkládání záznamu do B-stromu

Je-li kapacita daného uzlu naplněna, musí dojít k rozdělení uzlů:



Vkládání záznamu do B-stromu

Algoritmus:

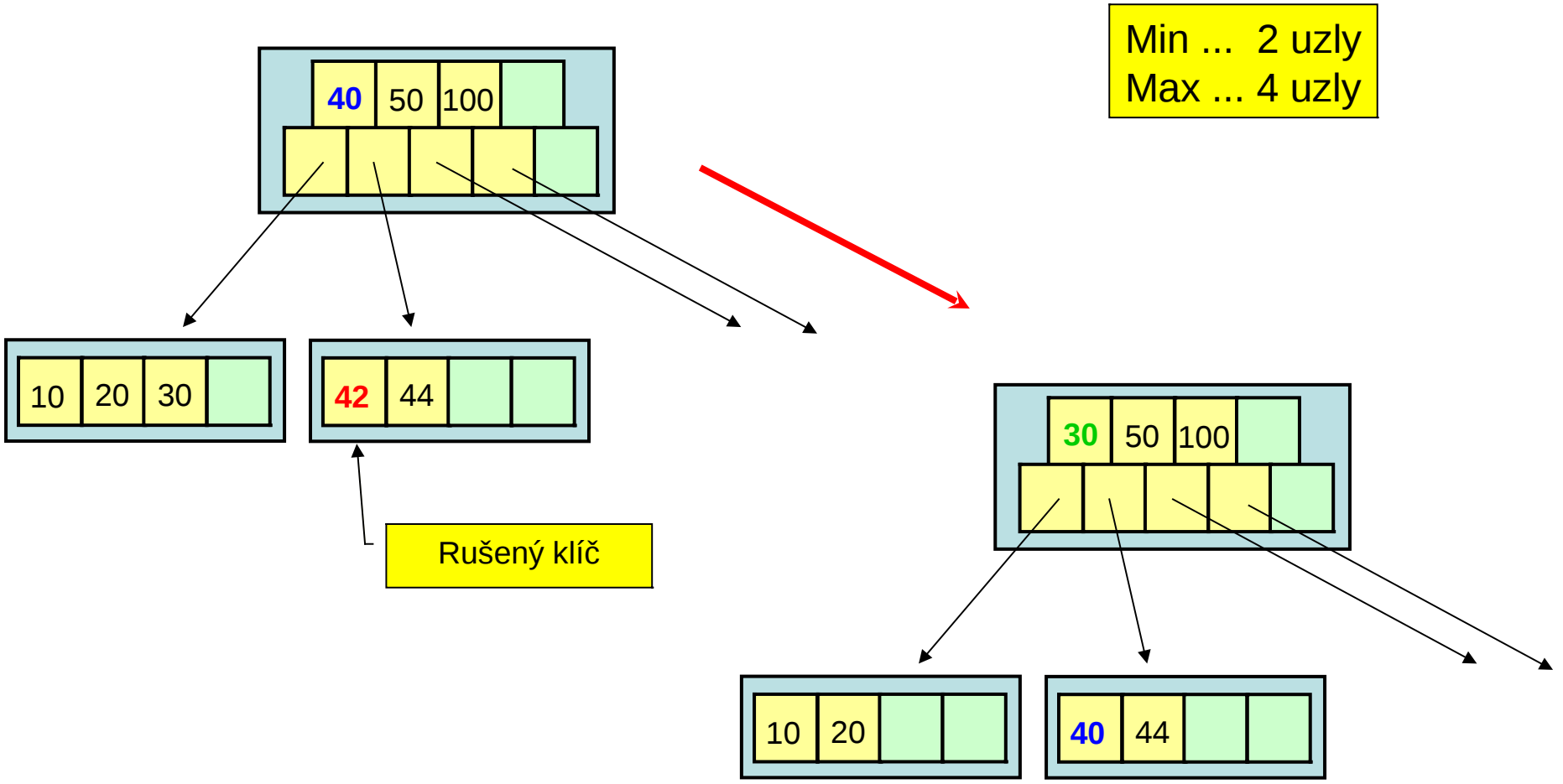
1. Pokud uzel, do něhož máme přidat nový záznam, obsahuje méně záznamů, než je jeho kapacita (maximální počet záznamů, který se „vejde“ do jednoho uzlu), vložíme nový záznam do tohoto uzlu
   při zachování uspořádání záznamů.
2. V opačném případě (uzel je naplněn na svou kapacitu), rozdělíme stávající uzel na dva uzly.
   a. Nalezneme separační záznam jako medián množiny klíčů, která je tvořena klíči záznamů existujících v děleném uzlu plus klíče vkládaného uzlu.
   b. Záznamy s klíči menšími než klíč separačního záznamu necháme v původním uzlu, záznamy s klíčem větším než klíč separačního záznamu uložíme do nového uzlu (zařazeného napravo od původního uzlu).
   c. Separační záznam vložíme do rodičovského uzlu, který se zase může rozlomit, pokud jeho kapacita již byla naplněna. Pokud uzel nemá rodiče (t.j. byl kořenem B-stromu), vytvoříme nový kořen nad tímto uzlem (dojde ke zvětšení hloubky stromu).

Rušení záznamu v listu B-stromu

- Rušení záznamu v **listu** B-stromu je jednodušší než rušení záznamu v nelistovém uzlu.
- Pokud po zrušení záznamu klesne počet záznamů pod minimální kapacitu a sourozenecký uzel má více záznamů, než je minimální kapacita uzlu, **přesuneme** záznam ze sourozeneckého uzlu.
- Klesne-li počet záznamů v obou sourozeneckých uzlech pod minimální kapacitu uzlu, uzly **spojíme**.

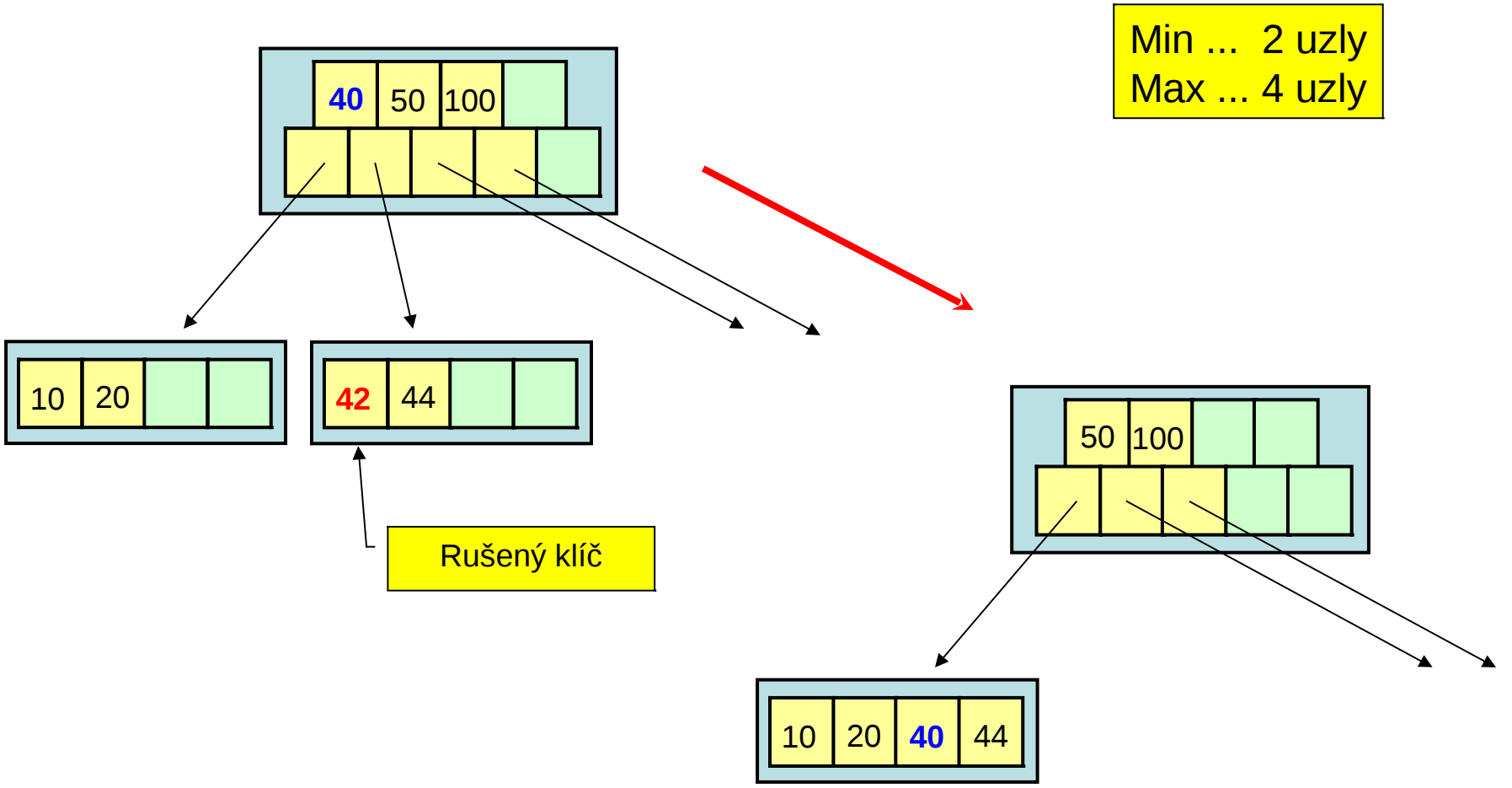
Rušení záznamu v listu B-stromu

Přesun uzlu

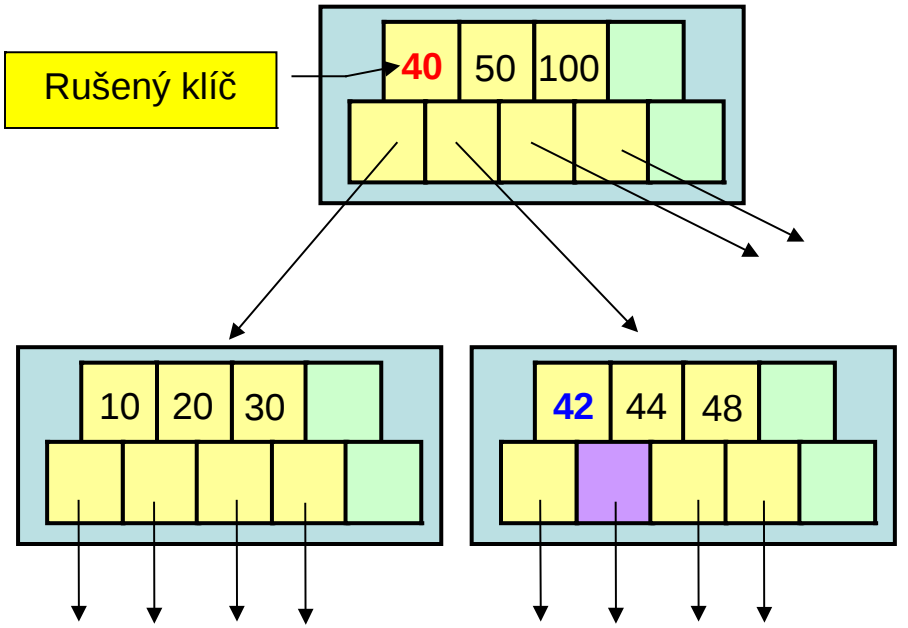


Rušení záznamu v listu B-stromu

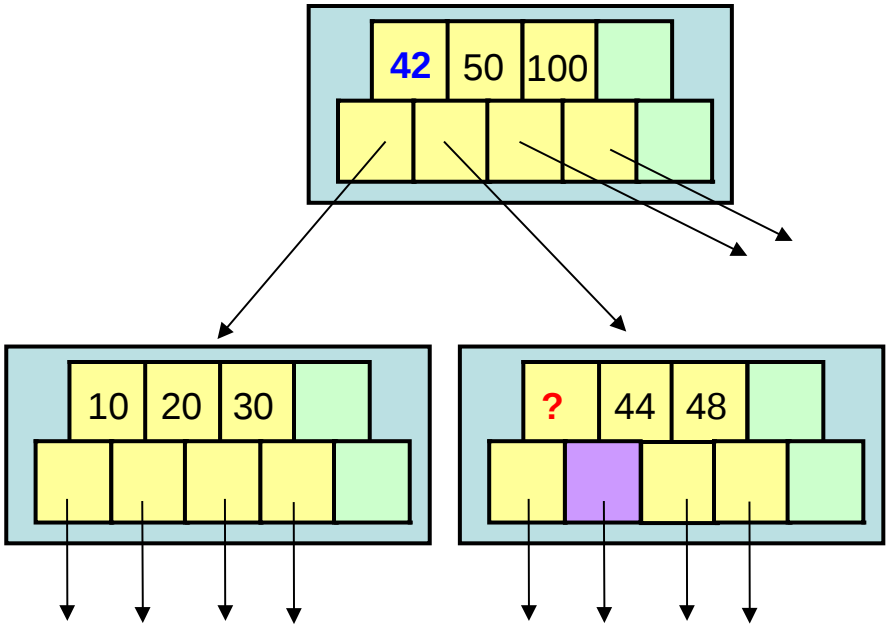
Spojení uzlů



Rušení záznamu v nelistovém uzlu B-stromu



Min ... 2 uzly
Max ... 4 uzly



Klíč označený **?** bude nejmenší klíč **fialového** podstromu.

Může následovat:
- spojení uzlů
- přesun od sourozence

Rušení záznamu v nelistovém uzlu B-stromu

- Tento přístup znamená, že při rušení uzlu smažeme rušený klíč a poté uvádíme strom znovu do vyváženého stavu.
- Není to jediný možný algoritmus, existují i jiné.

Prostorové indexační techniky

Zdeněk Kouba

Geografické informační systémy

- Data strukturovaná
  - Relační databáze
  - Dotazy SQL
- Data nestrukturovaná
  - Mapové podklady – rastrová data
  - Geometrické objekty – vektorová data
  - Geometrické závislosti
  - Prostorové dotazy

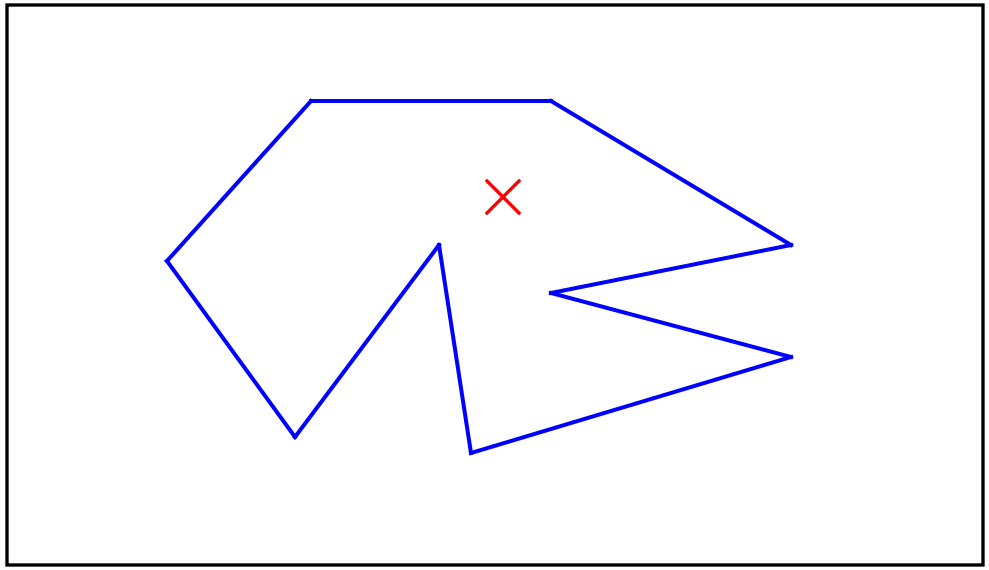
Hlavní typy prostorových dotazů

- **Dotaz na úplnou shodu** – vrátí všechny shodné objekty
- **Dotaz na bod** – nalézt všechny objekty o takové, že daný bod b leží uvnitř objektu o
- **Dotaz na oblast** – k dané oblasti (typicky polygonu) P nalézt všechny objekty o takové, že o má s P neprázdný průnik
- **Dotaz na okolí oblasti** – k dané oblasti P najít množinu objektů o takových, že $P \subseteq o$
- **Dotaz na obsah oblasti** – k dané oblasti P najít množinu objektů o takových, že $o \subseteq P$

Detaily: Jaroslav Pokorný: Prostorové datové struktury, GIS Ostrava ’99, Technická univerzita Ostrava, 1999

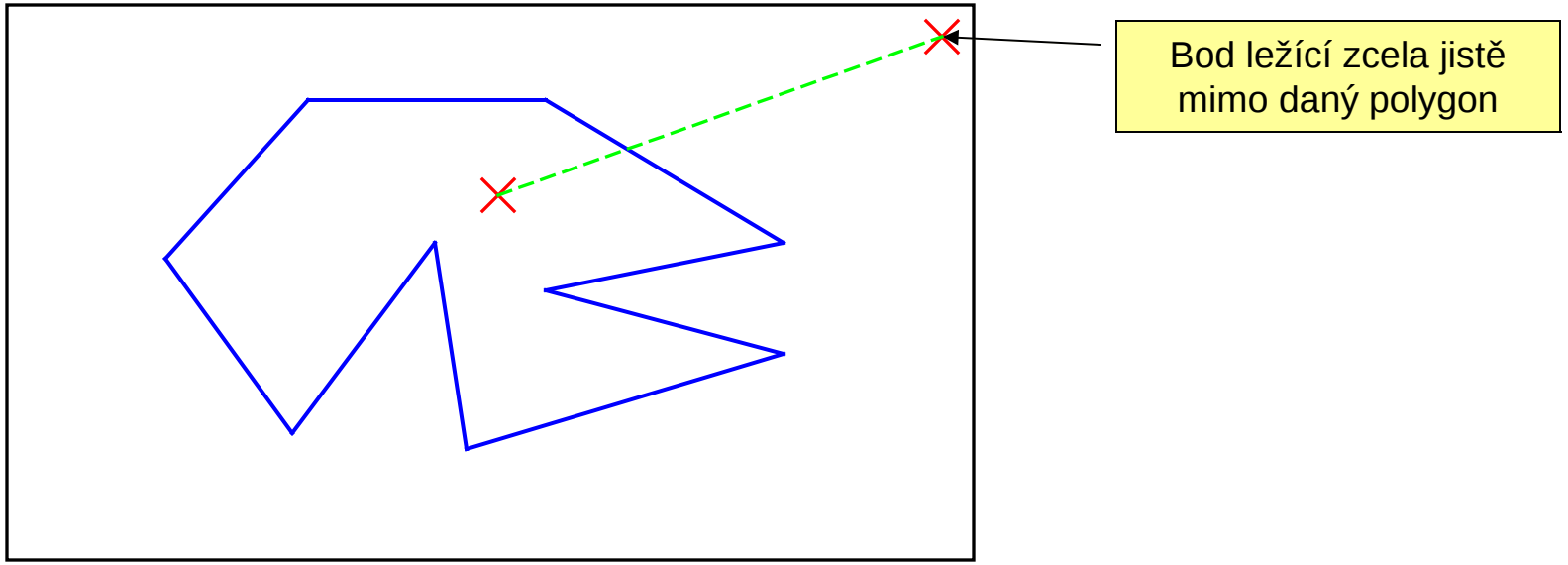
Prostorové dotazy

- Leží daný bod v daném polygonu?
- Efektivní algoritmus



Prostorové dotazy

- Leží daný bod v daném polygonu?
- Efektivní algoritmus



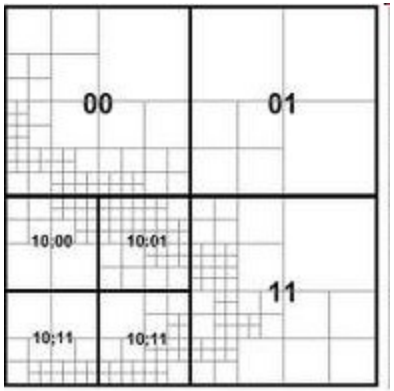
- Lichý počet průsečíků s hranicí polygonu => bod leží uvnitř polygonu
- Sudý počet – “ – => bod leží vně polygonu

Jak prostorové dotazy urychlit?

Prostorová indexace

Reprezentace prostoru

- Možnost: rozdělení prostoru do disjunktních buněk, objekty umistěny v těchto buňkách
- Hierarchie buněk: 4 - stromy (quad-tree) pro 2D, 8 – stromy pro 3D

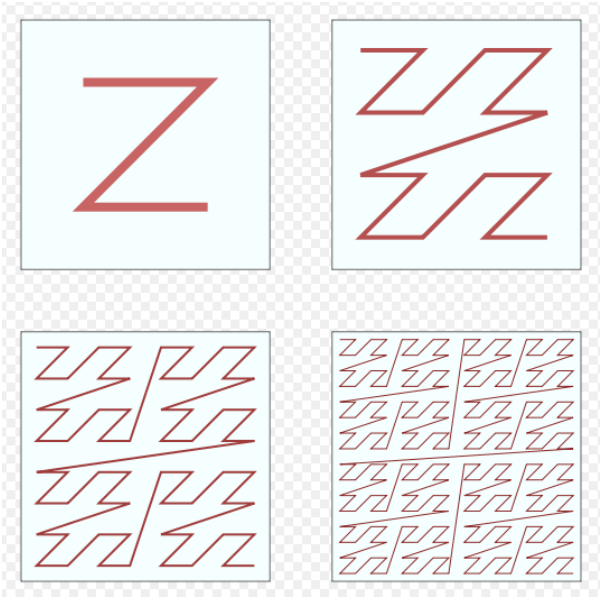


Reprezentace prostoru

- Číslování buněk: využití křivek vyplňujících prostor
  - princip Hilbertovy křivky
  - Mortonův rozklad (Z-order curve) – viz obrázek

    - výhoda: zachovává sousedství (sousední buňky na křivce = sousední buňky v prostoru)
    - nevýhoda: výpočetní složitost

Z curve:



http://en.wikipedia.org/wiki/File:Four-level_Z.svg

Reprezentace prostoru

- Z curve
  - číslování – prolnutí bitů souřadnic x a y v binárním vyjádření

	x: 0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
y: 0 000	000000	000001	000100	000101	010000	010001	010100	010101
1 001	000010	000011	000110	000111	010010	010011	010110	010111
2 010	001000	001001	001100	001101	011000	011001	011100	011101
3 011	001010	001011	001110	001111	011010	011011	011110	011111
4 100	100000	100001	100100	100101	110000	110001	110100	110101
5 101	100010	100011	100110	100111	110010	110011	110110	110111
6 110	101000	101001	101100	101101	111000	111001	111100	111101
7 111	101010	101011	101110	101111	111010	111011	111110	111111

http://en.wikipedia.org/wiki/File:Z-curve.svg

Prostorová indexace

- Vyplňující křivka (Z curve) nám n-dimenzionální problém převádí na jednorozměrný
- Pro jednorozměrné problémy máme efektivní indexační techniky, hashování, ...

Prostorová indexace

- Vyplňující křivka (Z curve) převádí n-dimenzionální problém na jednorozměrný
- Pro jednorozměrné problémy máme efektivní indexační techniky (B-stromy), hashování, ...

Máme tedy prostorovou indexaci vyřešenu?

Prostorová indexace

- Vyplňující křivka (Z curve) převádí n-dimenzionální problém na jednorozměrný
- Pro jednorozměrné problémy máme efektivní indexační techniky, hashování, ...

Máme tedy prostorovou indexaci vyřešenu?

- Držíme-li všechny prostorové objekty v operační paměti, pak ano (malé úlohy).
- Jsou-li prostorové objekty uloženy na disku, pak tento přístup nezohledňuje potřebu minimalizovat počet přístupů na disk.

Co tedy potřebujeme

Prostorová indexace

Co tedy potřebujeme

Buňky sousedící v prostoru by měly být uloženy v téže stránce externí paměti.

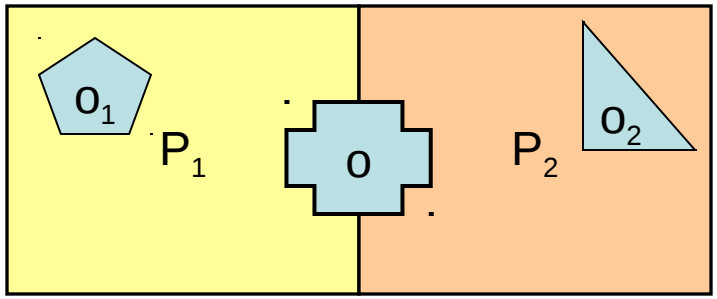
Prostorová indexace

Nepřekrývající se oblasti:

- Prostor P je rozdělen do navzájem disjunktních podprostorů.
- Tyto podprostory mohou být dále rekurzivně děleny do disjunktních podprostorů
  => vznikne hierarchická struktura

Nechť je prostor **P** rozdělen do dvou disjunktních podprostorů $\mathbf{P_1}$ a $\mathbf{P_2}$.

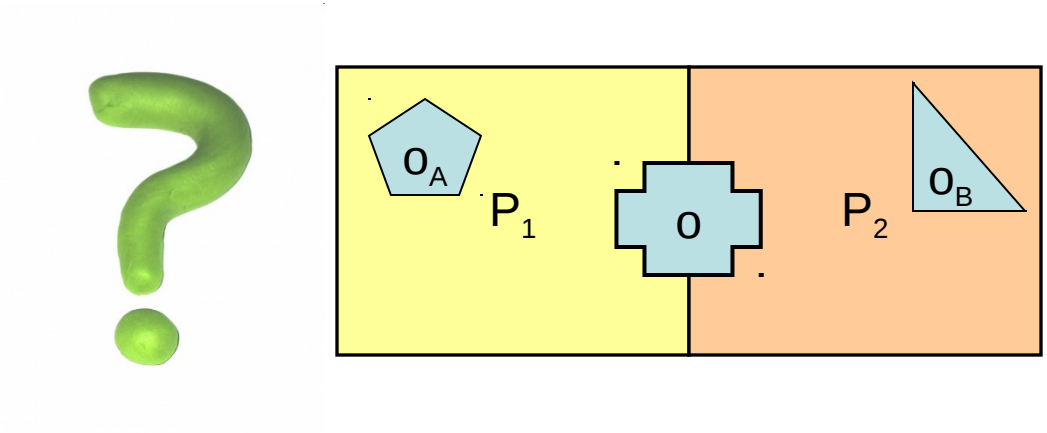
Co když máme objekt **o** takový, že má neprázdný průnik jak s $\mathbf{P_1}$ tak s $\mathbf{P_2}$?



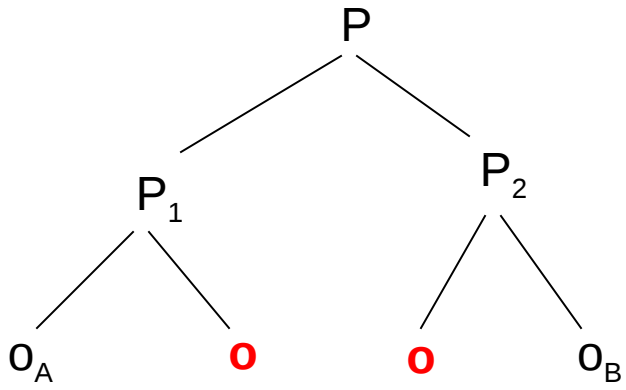
Prostorová indexace

Nepřekrývající se oblasti:

Co když máme objekt **o** takový, že má neprázdný průnik jak s P_1 tak s P_2?



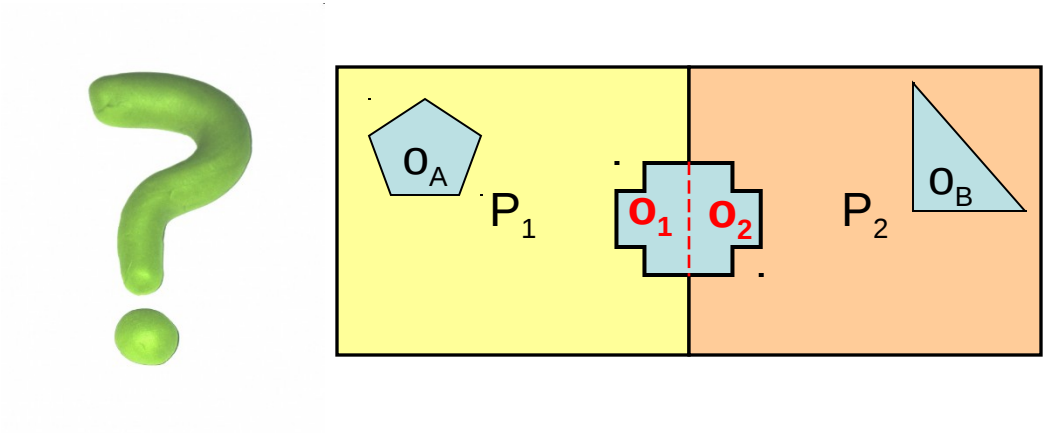
1. možnost: duplikace objektů:



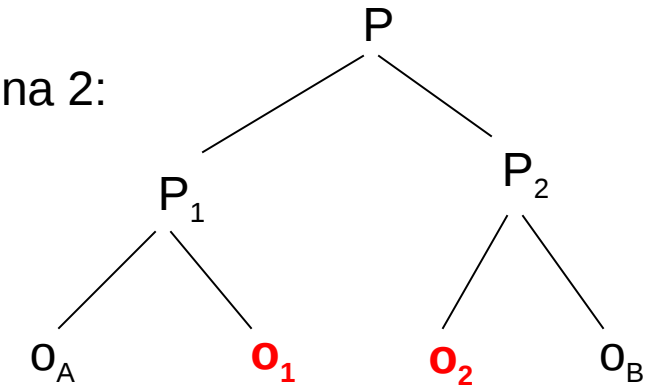
Prostorová indexace

Nepřekrývající se oblasti:

Co když máme objekt **o** takový, že má neprázdný průnik jak s P_1 tak s P_2?



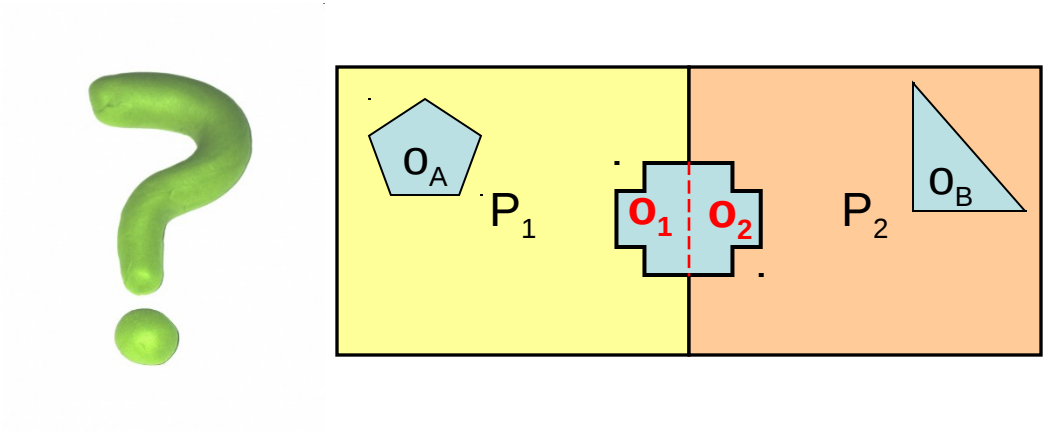
2. možnost: rozdělení objektu o na 2:
(clipping)



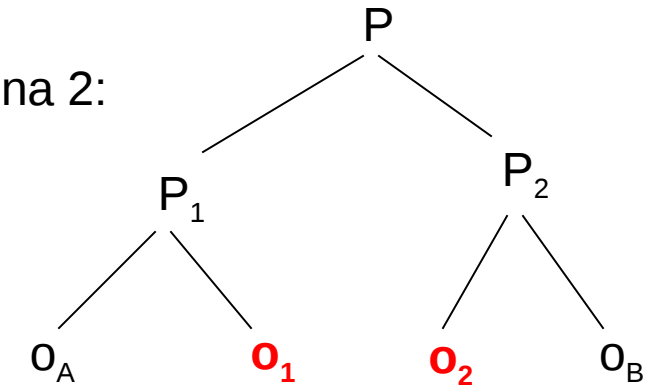
Prostorová indexace

Nepřekrývající se oblasti:

Co když máme objekt **o** takový, že má neprázdný průnik jak s P_1 tak s P_2?

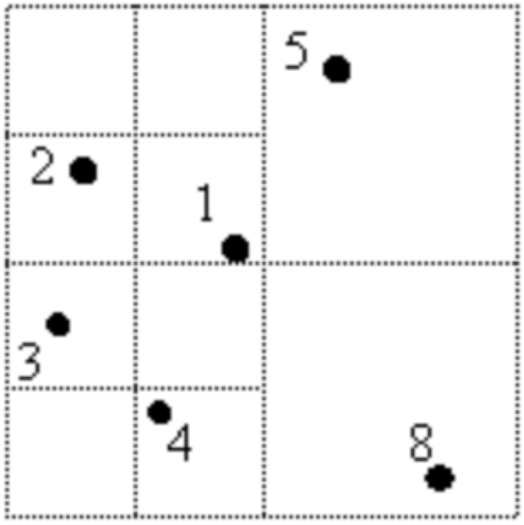


2. možnost: rozdělení objektu o na 2:
(clipping)



Prostorové struktury pro indexaci bodů

4-stromy (quad-trees)

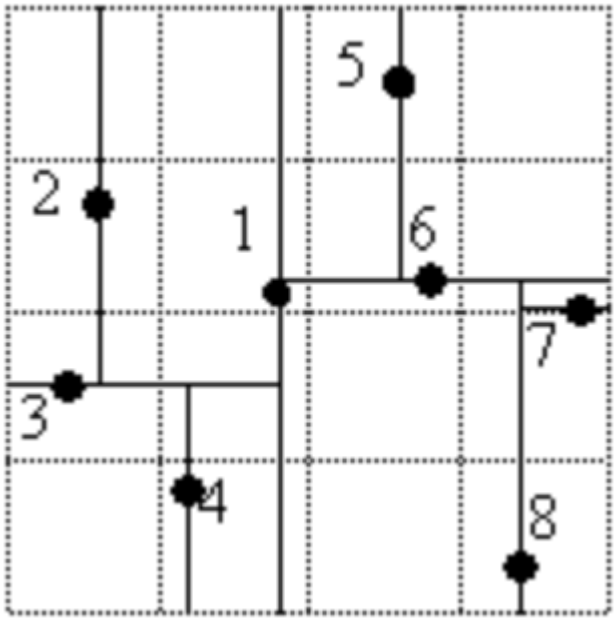


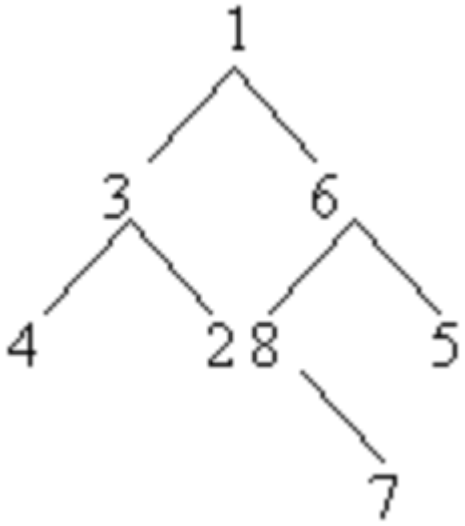
4-stromy jsou obecně nevyvážené.

Obrázek převzat z:
Jaroslav Pokorný: **Prostorové datové struktury a jejich použití k indexaci prostorových objektů**
http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2000/Sbornik/Pokorny/Referat.htm

Prostorové struktury pro indexaci bodů

k-d-stromy (k-d-trees)



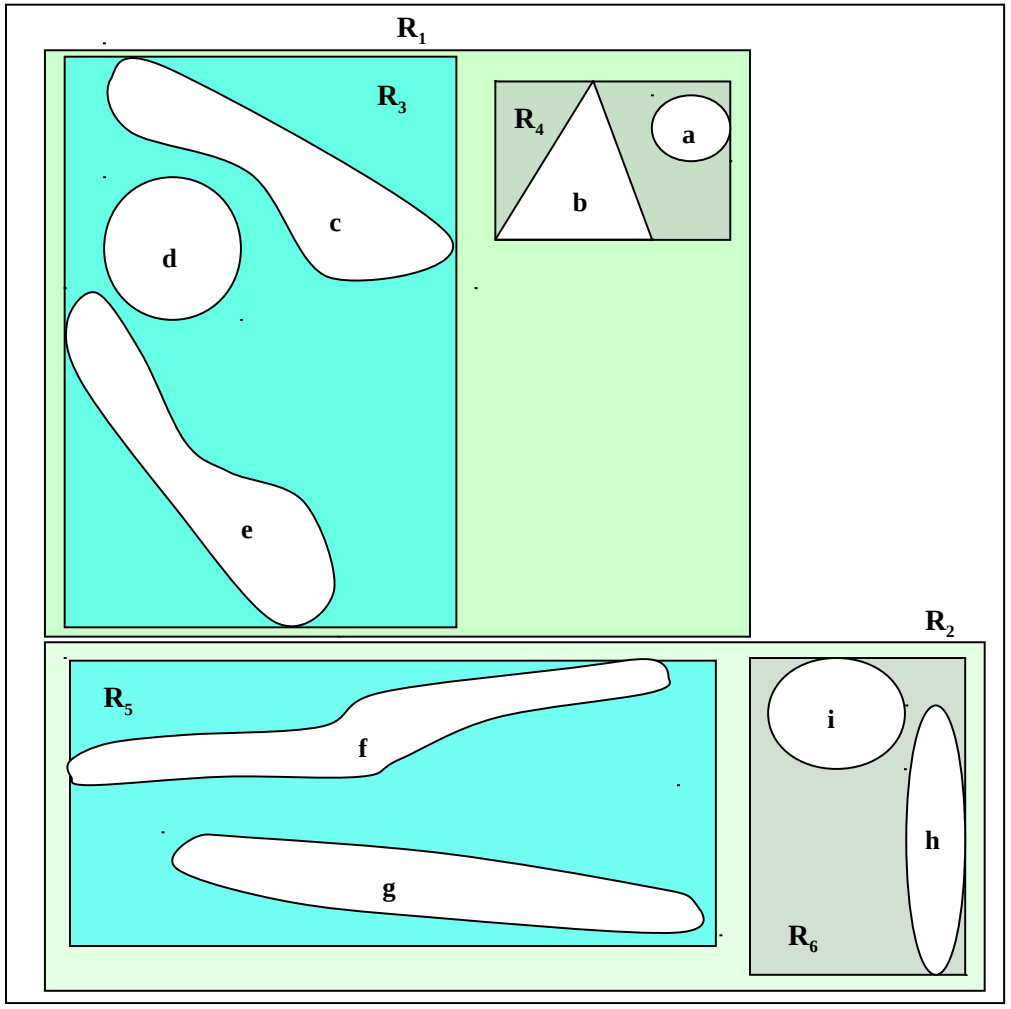


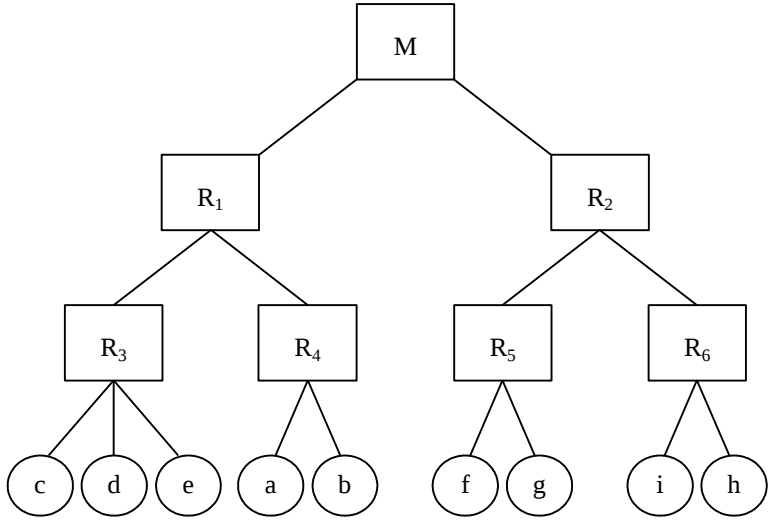
Ukázka 2-d stromu

k-d-stromy jsou obecně vyvážené.

Obrázek převzat z:
Jaroslav Pokorný: **Prostorové datové struktury a jejich použití k indexaci prostorových objektů**
http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2000/Sbornik/Pokorny/Referat.htm

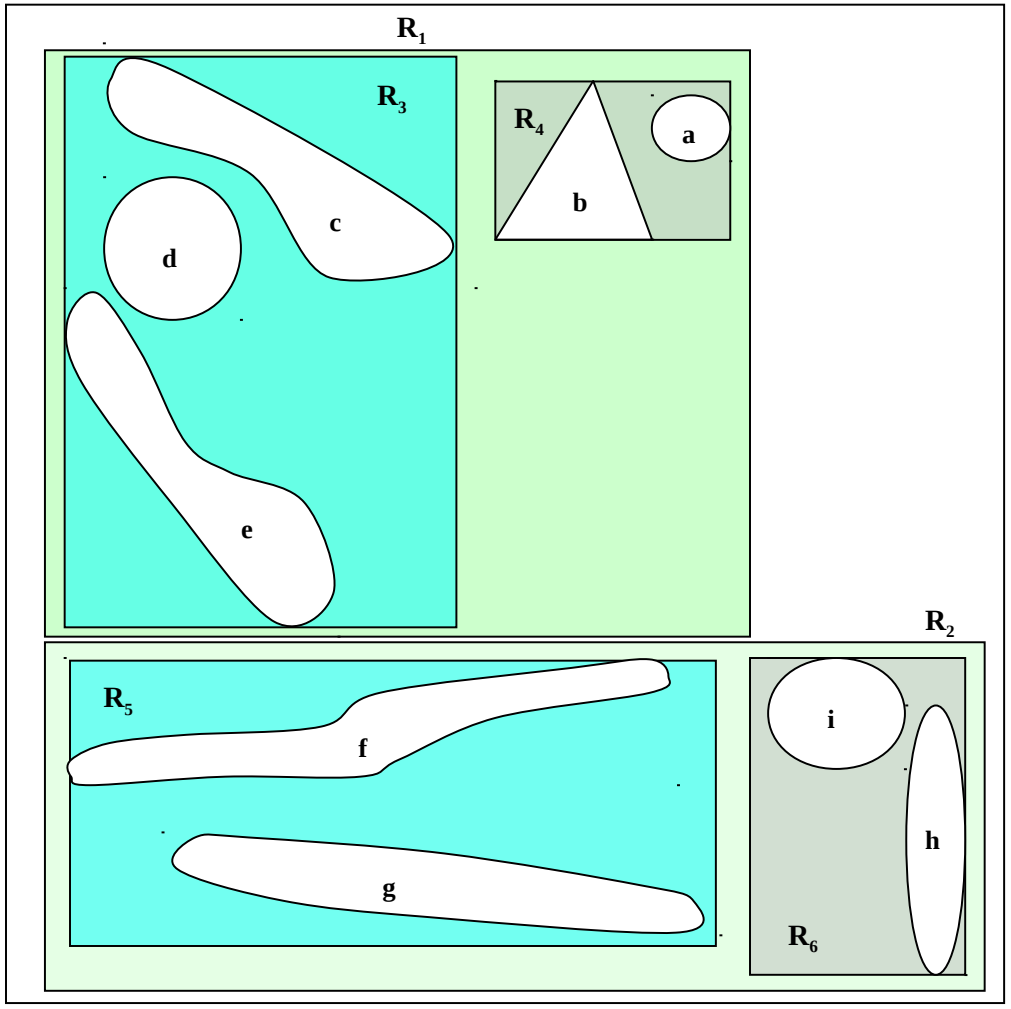
R - strom

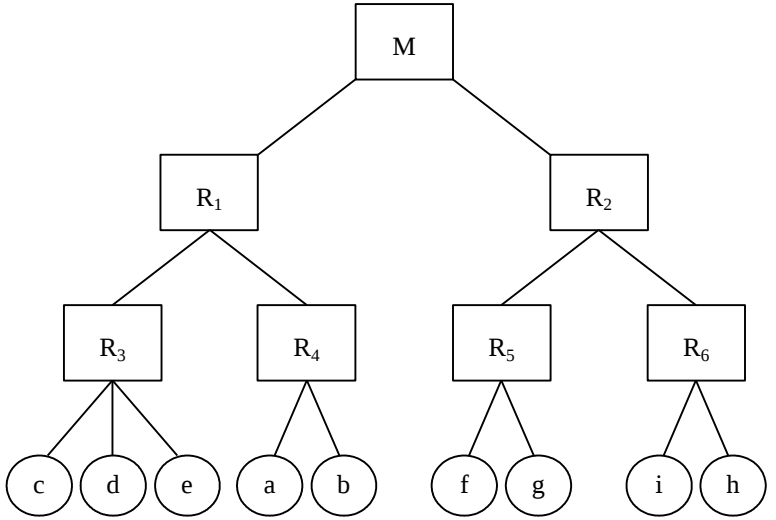




MBR – minimal bounding rectangle

R - strom





Heuristika: aby byl minimalizován „hluchý“ prostor, volí takový MBR, který má minimální plochu.

R - strom

MBR se mohou překrývat
Heuristika: aby byl minimalizován „hluchý“ prostor, volí takový MBR, který má minimální plochu.

Insert: možnost dělení uzlu (překročení max. kapacity uzlu)
objekt může být přidán do více uzlů
Delete: možnost slučování uzlů (pokles počtu uzlů pod minimální kapacitu uzlu)

R* - strom

Heuristika: aby byl minimalizován „hluchý“ prostor, volí takový MBR, který má minimální plochu + minimalizace obvodu + minimalizace překryvu

Výkon R* stromů prudce klesá se stoupající dimenzí

R^+ - strom

kompromis mezi R-trees a k-d-trees

MBR se nepřekrývají
Clipping nebo jsou jsou objekty vkládány do více listů stromu
Nepoužívá se minimální kapacita