

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**

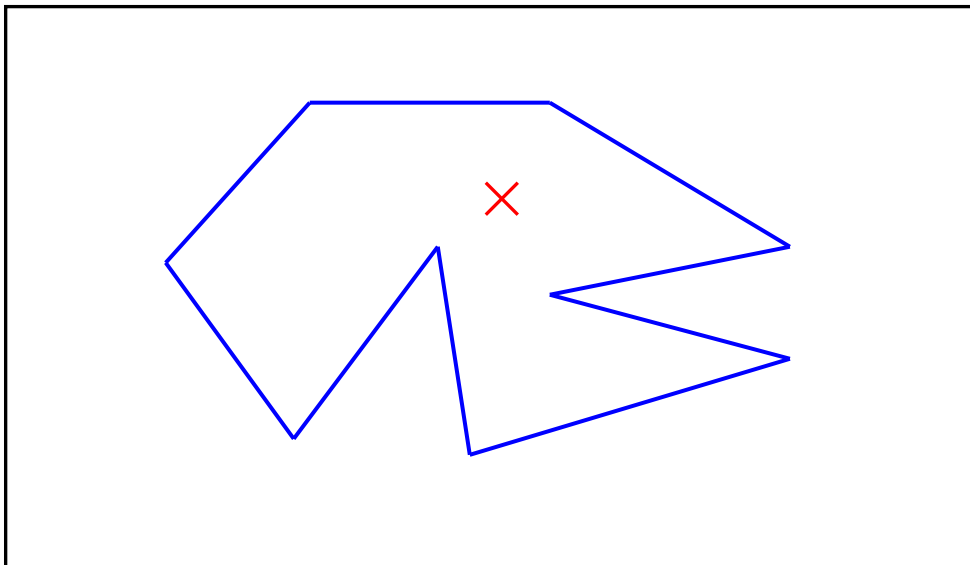
Hlavní typy prostorových dotazů

- **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

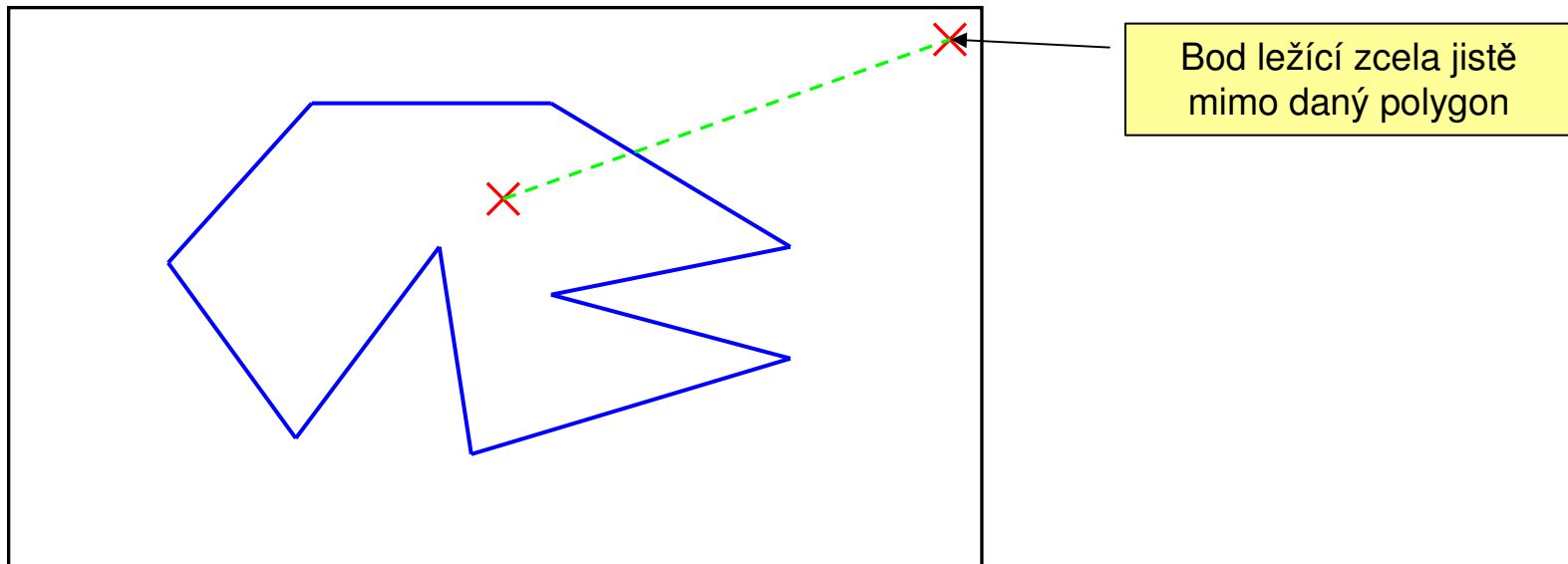
Prostorové dotazy

- Leží daný bod v daném polygonu?
- Efektivní algoritmus



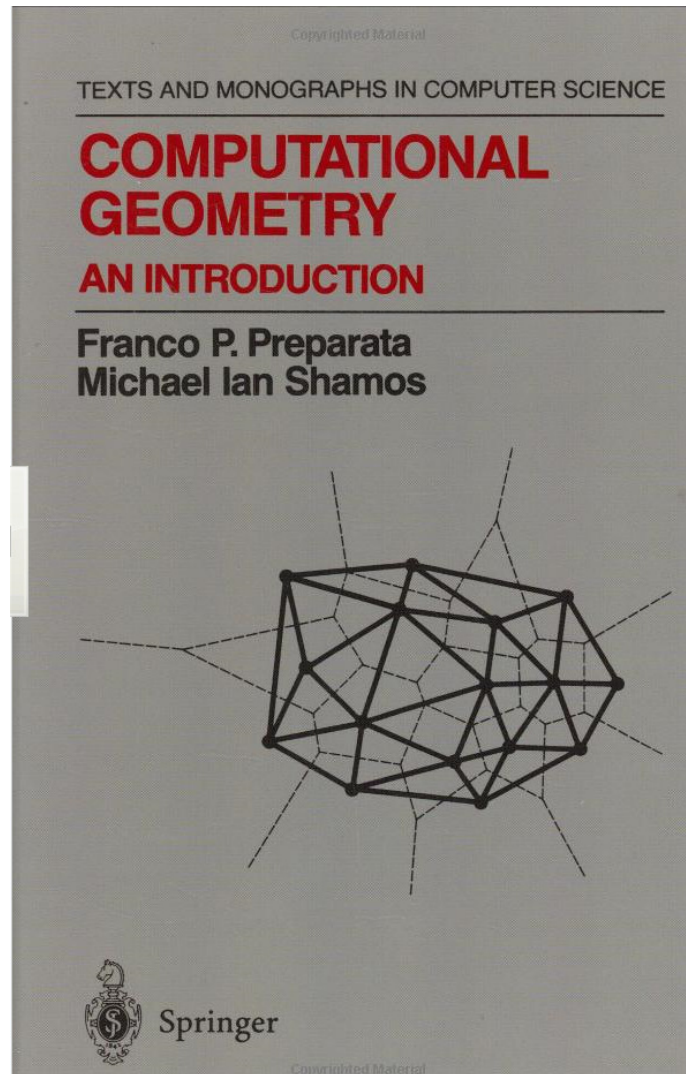
Prostorové dotazy

- Leží daný bod v daném polygonu?
- Efektivní algoritmus



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

Výpočetní geometrie



Jak prostorové dotazy urychlit?

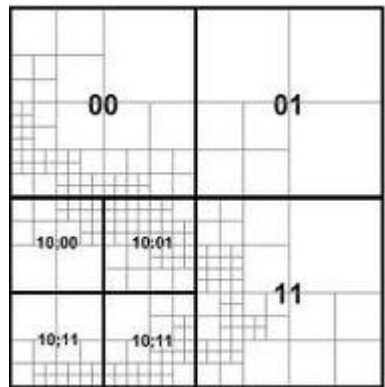


Prostorová indexace



Reprezentace prostoru

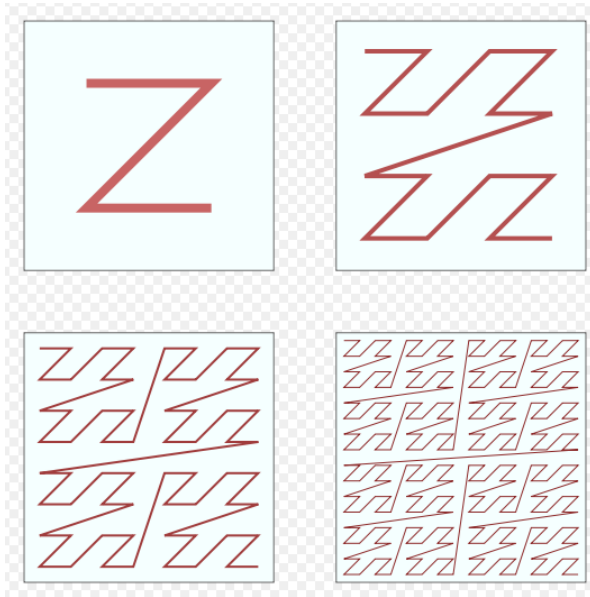
- Možnost: rozdělení prostoru do disjunktních buněk, objekty umístěny v těchto buňkách
- Hierarchie buněk: 4 - stromy (quad-tree) pro 2D, 8 – stromy pro 3D



Reprezentace prostoru

- Číslování buněk: využití křivek vyplňujících prostor
 - Z curve – viz obrázek
 - Hilbertova křivka
 - 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	1	2	3	4	5	6	7
		000	001	010	011	100	101	110	111
y: 0		000000	000001	000100	000101	010000	010001	010100	010101
1		000010	000011	000110	000111	010010	010011	010110	010111
2		001000	001001	001100	001101	011000	011001	011100	011101
3		001010	001011	001110	001111	011010	011011	011110	011111
4		100000	100001	100100	100101	110000	110001	110100	110101
5		100010	100011	100110	100111	110010	110011	110110	110111
6		101000	101001	101100	101101	111000	111001	111100	111101
7		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-dimensionální problém převádí na jednorozměrný
- Pro jednorozměrné problémy máme efektivní indexační techniky, hashování, ...

Prostorová indexace

- Vyplňující křivka (Z curve) převádí n-dimensionální 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-dimensionální 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.



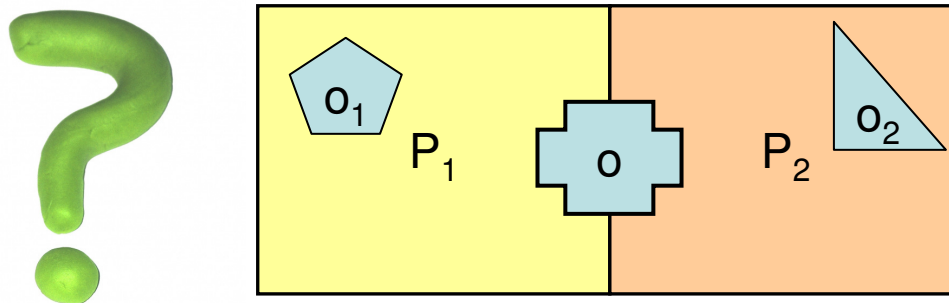
Prostorová indexace

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 P_1 a P_2 .

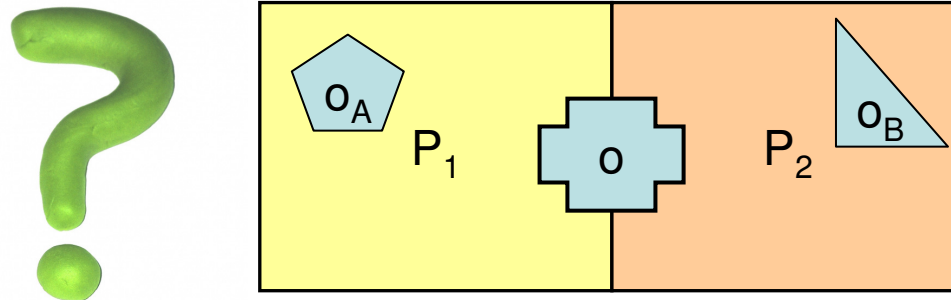
Co když máme objekt o takový, že má neprázdný průnik jak s P_1 tak s P_2 ?



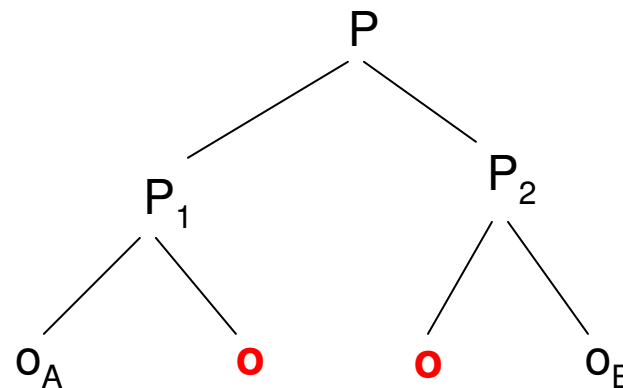
Prostorová indexace

Nepřekrývající se oblasti:

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



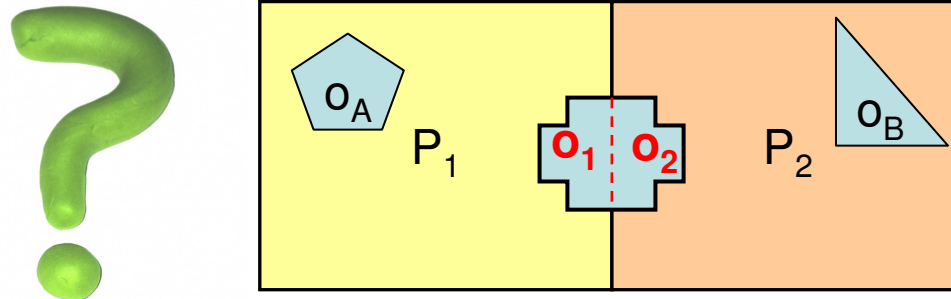
1. možnost: duplikace objektů:



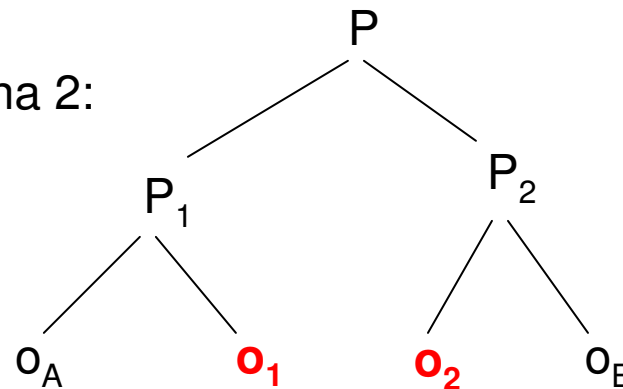
Prostorová indexace

Nepřekrývající se oblasti:

Co když máme objekt **o** takový, že má neprázdný průnik jak s **P₁** tak s **P₂**?



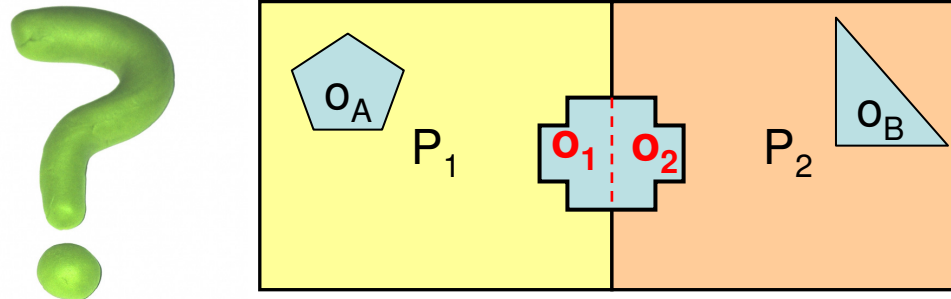
2. možnost: rozdělení objektu o na 2:
(clipping)



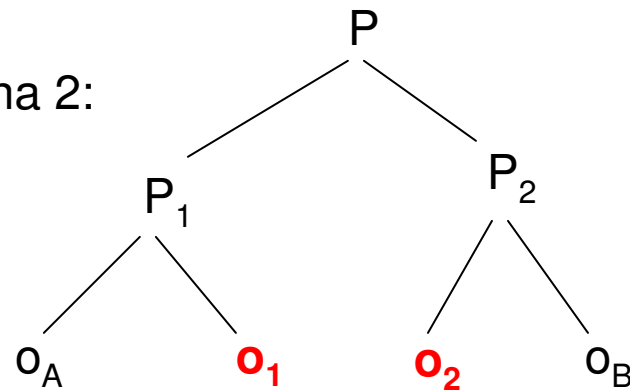
Prostorová indexace

Nepřekrývající se oblasti:

Co když máme objekt **o** takový, že má neprázdný průnik jak s **P₁** tak s **P₂**?

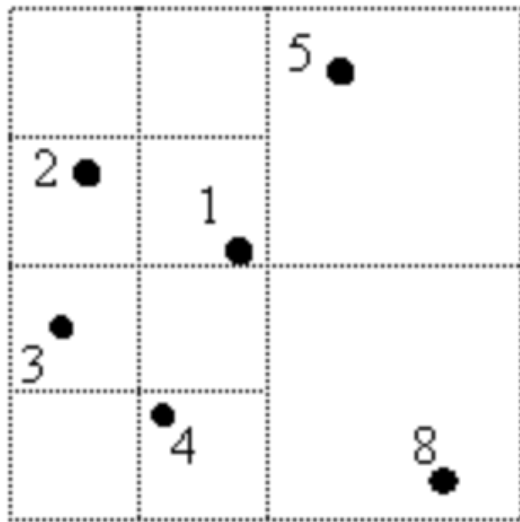


2. možnost: rozdělení objektu o na 2:
(clipping)



Prostorové struktury pro indexaci bodů

4-stromy (quad-trees)



4-stromy jsou obecně nevyvážené.

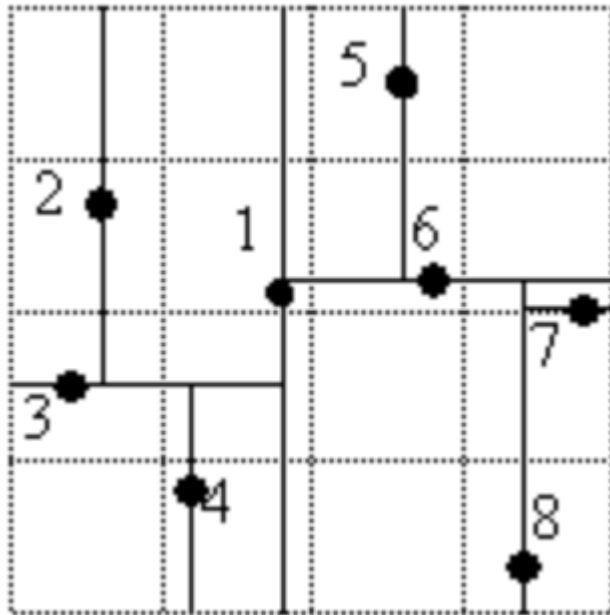
Obrázek převzat z:

Jaroslav Pokorný: Prostorové datové struktury a jejich použití k indexaci prostorových objektů

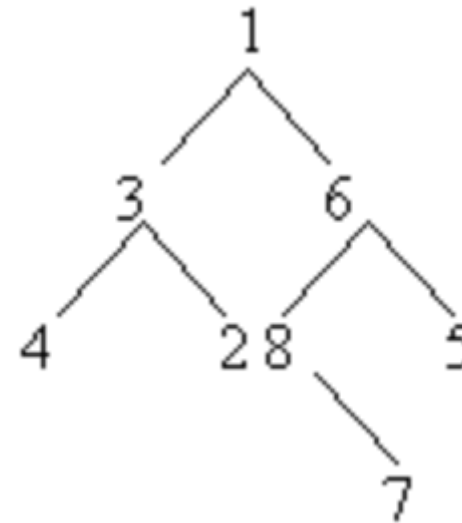
http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2000/Sbornik/Pokorny/Referat.htm

Prostorové struktury pro indexaci bodů

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



Ukázka 2-d stromu



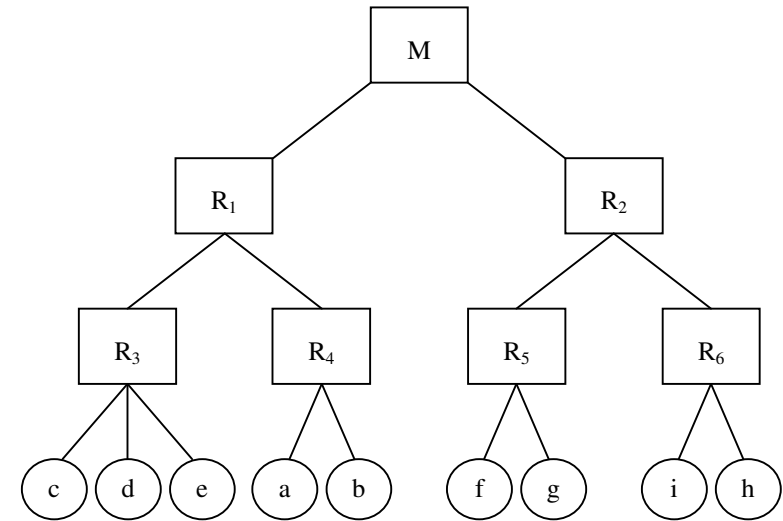
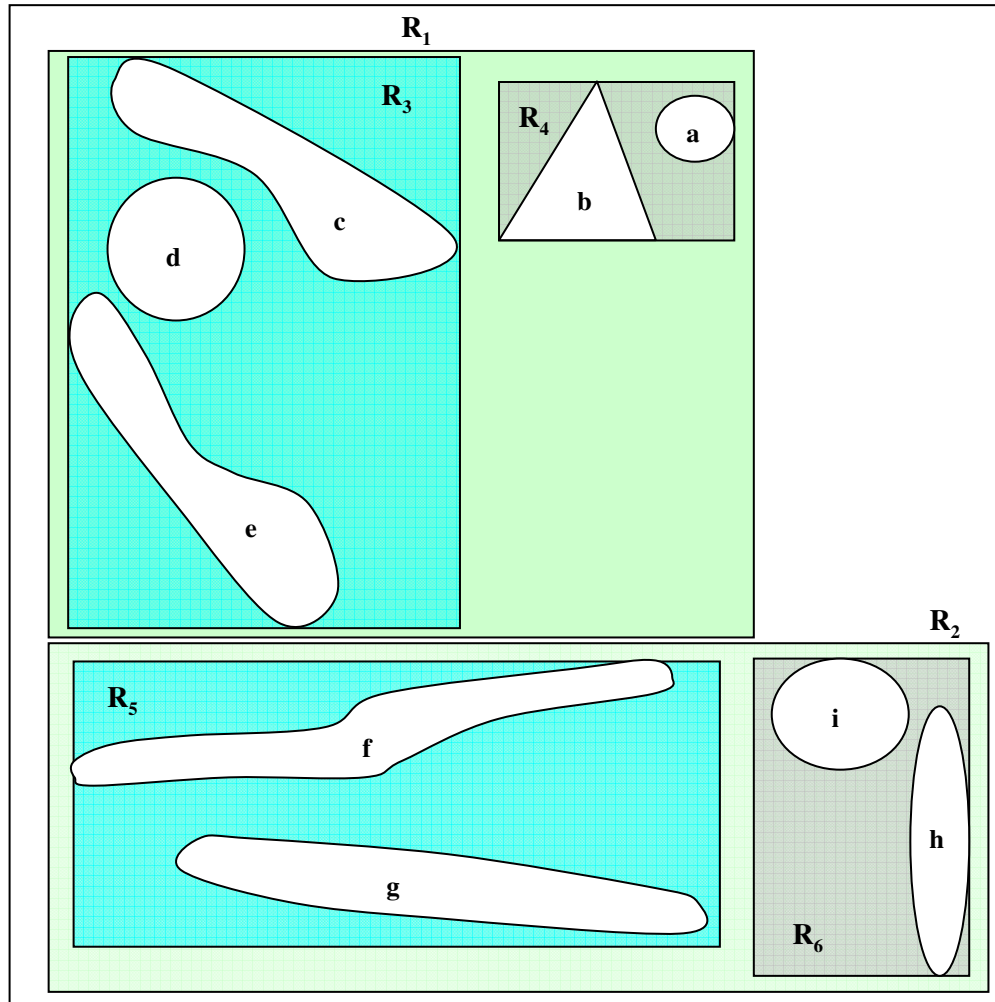
k-d-stromy jsou obecně vyvážené.

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-stromy – Antonin Guttman



Guttman, A: R-trees: a dynamic index structure for spatial searching.
Proc. Of SIGMOD Int. Conf. On Management of Data, 1984, pp. 47-54

Tehdy: PhD student, **University of California, Berkeley**
Dnes: Software Eng., **Oracle Corp.**

R-stromy

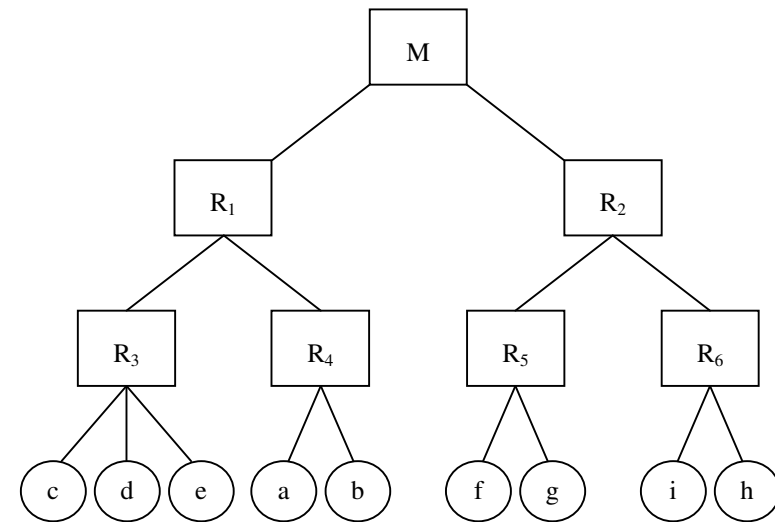
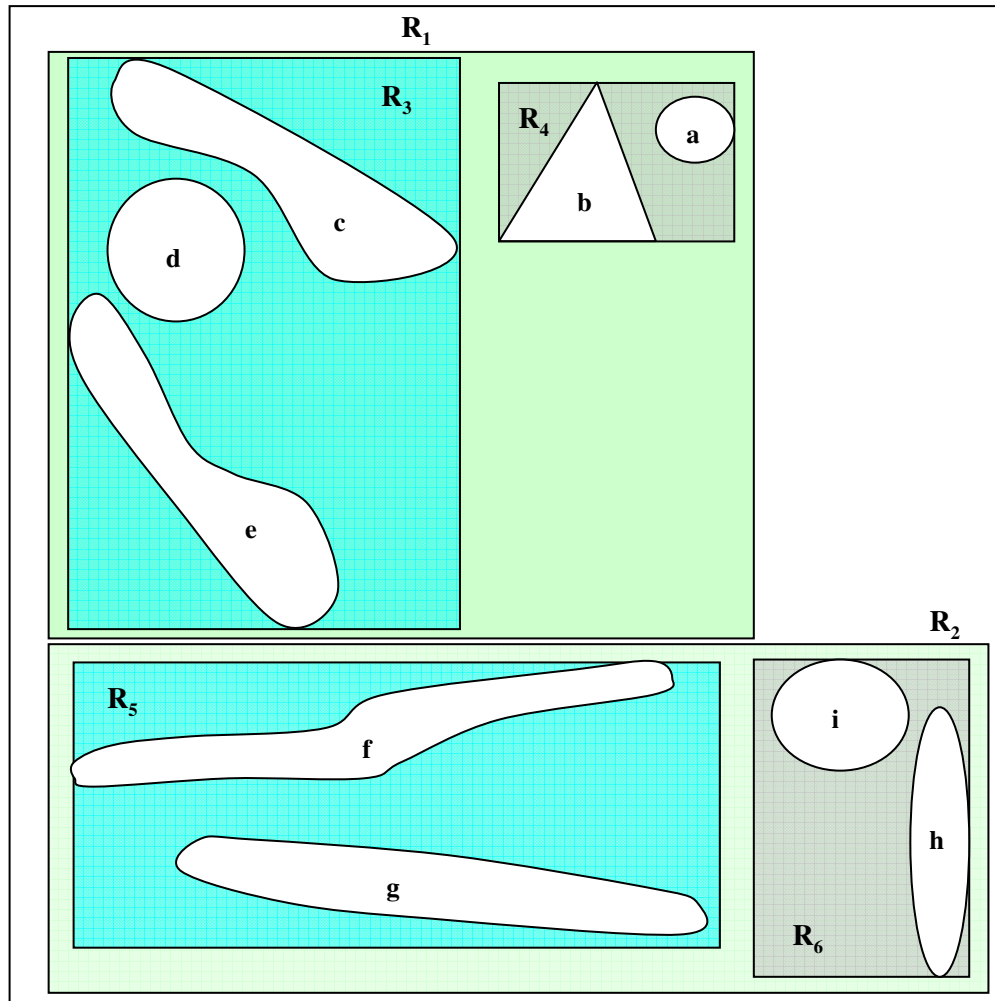
Odkaz na

- snadno dostupnou
- dobře napsanou literaturu

R-stromy

```
//=====
// Процедура поиска в R-дереве по области S
// Параметры:
// V - текущая вершина (первоначально это корень)
// S - область поиска
// Res - множество результатов поиска
//=====
ПОИСК(V, S, Res)
    [1] Если V не является листом, то
        Проверить все записи V', находящиеся в узле V
        Если MBR(V') пересекается с S, то
            Вызвать ПОИСК(V', S, Res)
    [2] Если V является листом, то
        Проверить все записи O, находящиеся в узле V
        Если MBR(O) пересекается с S, то
            Добавить запись O в множество Res
Конец ПОИСК
```

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 dimensí

R⁺ - strom

MBR se nepřekrývají

Clipping

Nepoužívá se minimální kapacita