Datové struktury pro prostorové vyhledávání

© 1998-2011 Josef Pelikán

CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

http://cgg.mff.cuni.cz/~pepca/

Aplikační oblasti

- **geografické informační systémy** (GIS)
  - plošné, liniové i bodové objekty
  - databáze mohou obsahovat **10^4** až **10^7** objektů
- **zpracování a analýza obrazu, rozpoznávání**
- **počítačová grafika, hry**
  - urychlování 2D i 3D algoritmů (sledování paprsku)
- **průmysl, CAD**
  - návrh VLSI, rozmisťování součástek, kontrola kolizí

Příklady elementárních úloh

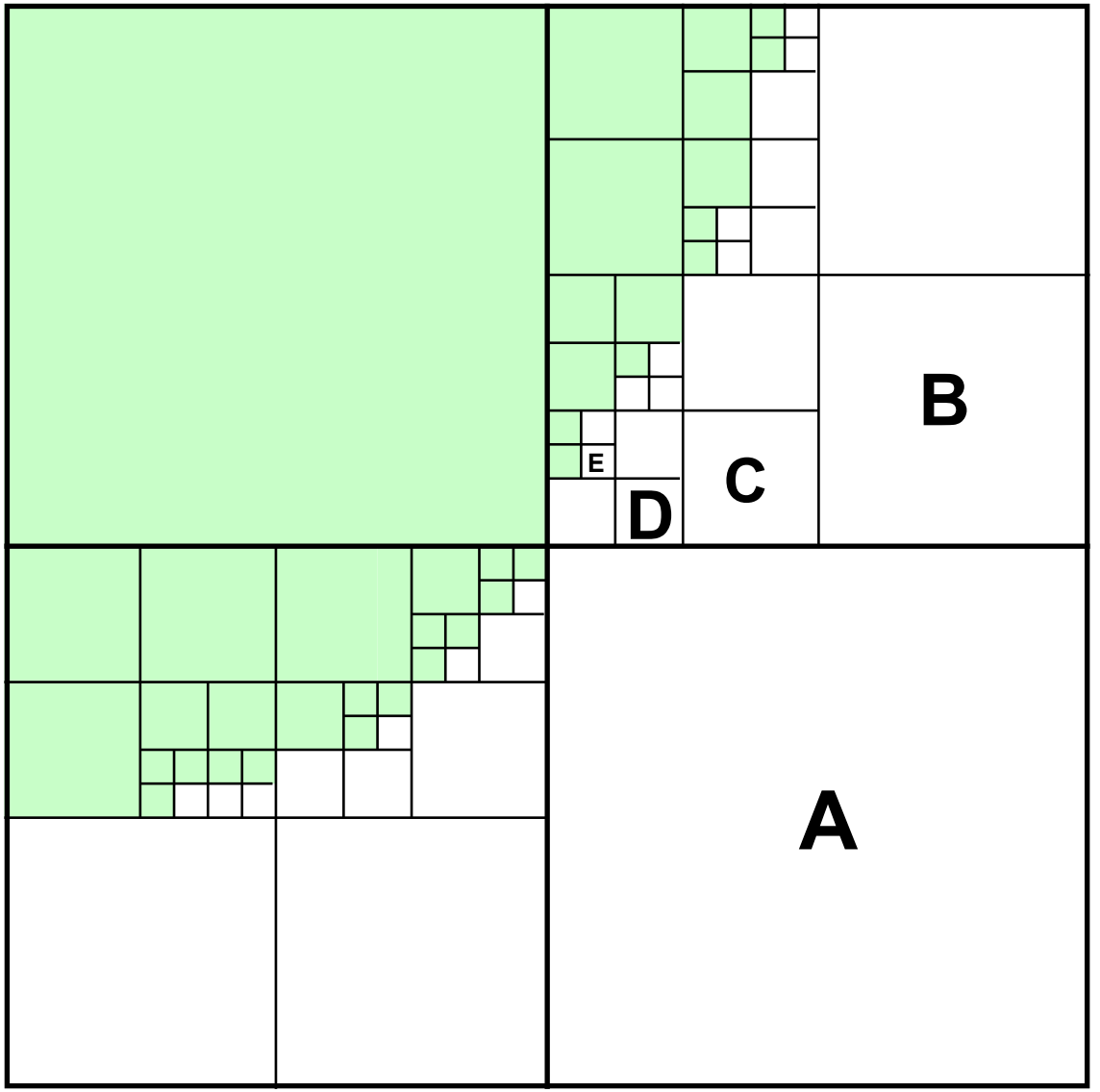
- **lokalizace bodu** v 2-3D síti
  – hledání plošného objektu, kterému patří daný bod
- hledání **nejbližších N bodů** od daného středu
  – globální varianta: hledání nejbližší dvojice bodů
- **průsečíky dvou křivek** (lomených čar)
  – hledání kolizí v množině křivek
- hledání **nejbližších bodových objektů** k dané křivce

Příklady elementárních úloh

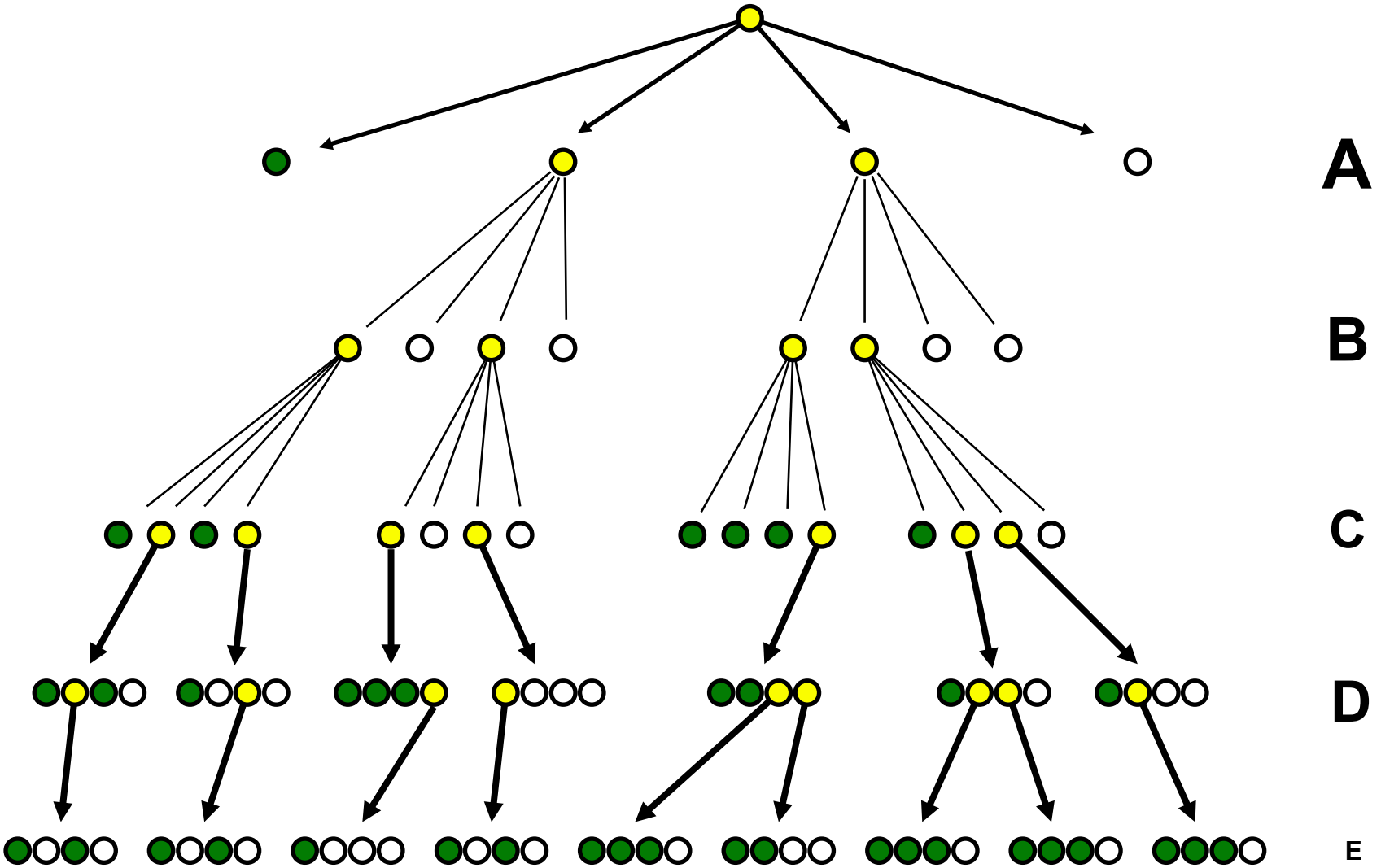
- **intervalové dotazy** v 2-3D prostoru
- **množinové operace** nad mapovými objekty
  – oblasti, linie, bodové objekty
- **testy kolizí (interferencí)**, minimální vzdálenosti mezi plošnými objekty (VLSI)
- hledání nejbližšího **průsečíku paprsku** s 3D scénou
- zpracování objektů **podle vzdálenosti** od středu

„Region quadtree”

- **reprezentace <u>plošných oblastí</u> v rovině**
- **dělení se provádí přesně <u>v polovině</u>**
- **informace o oblastech nesou pouze <u>listy</u>**

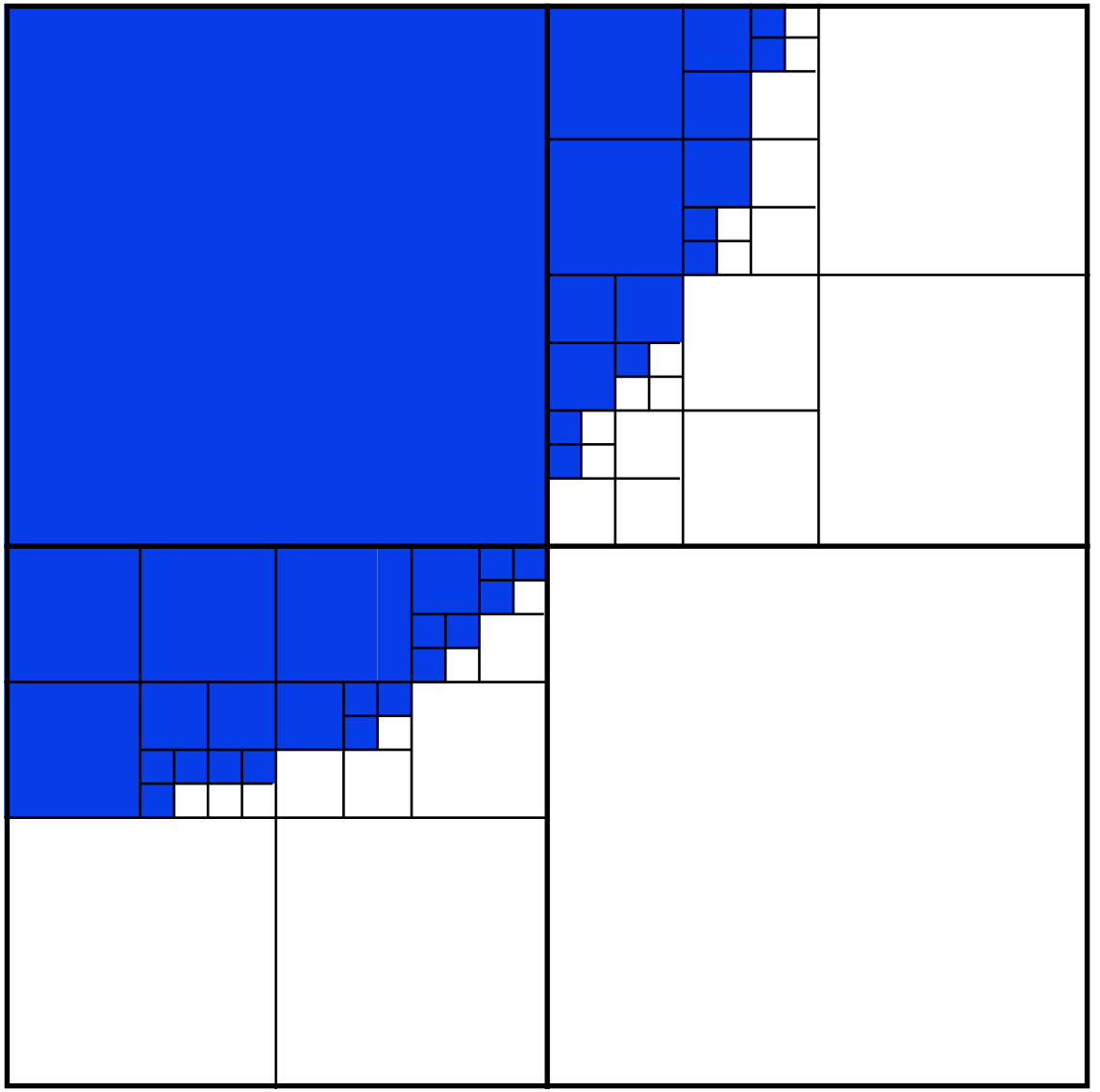


„Region quadtree”



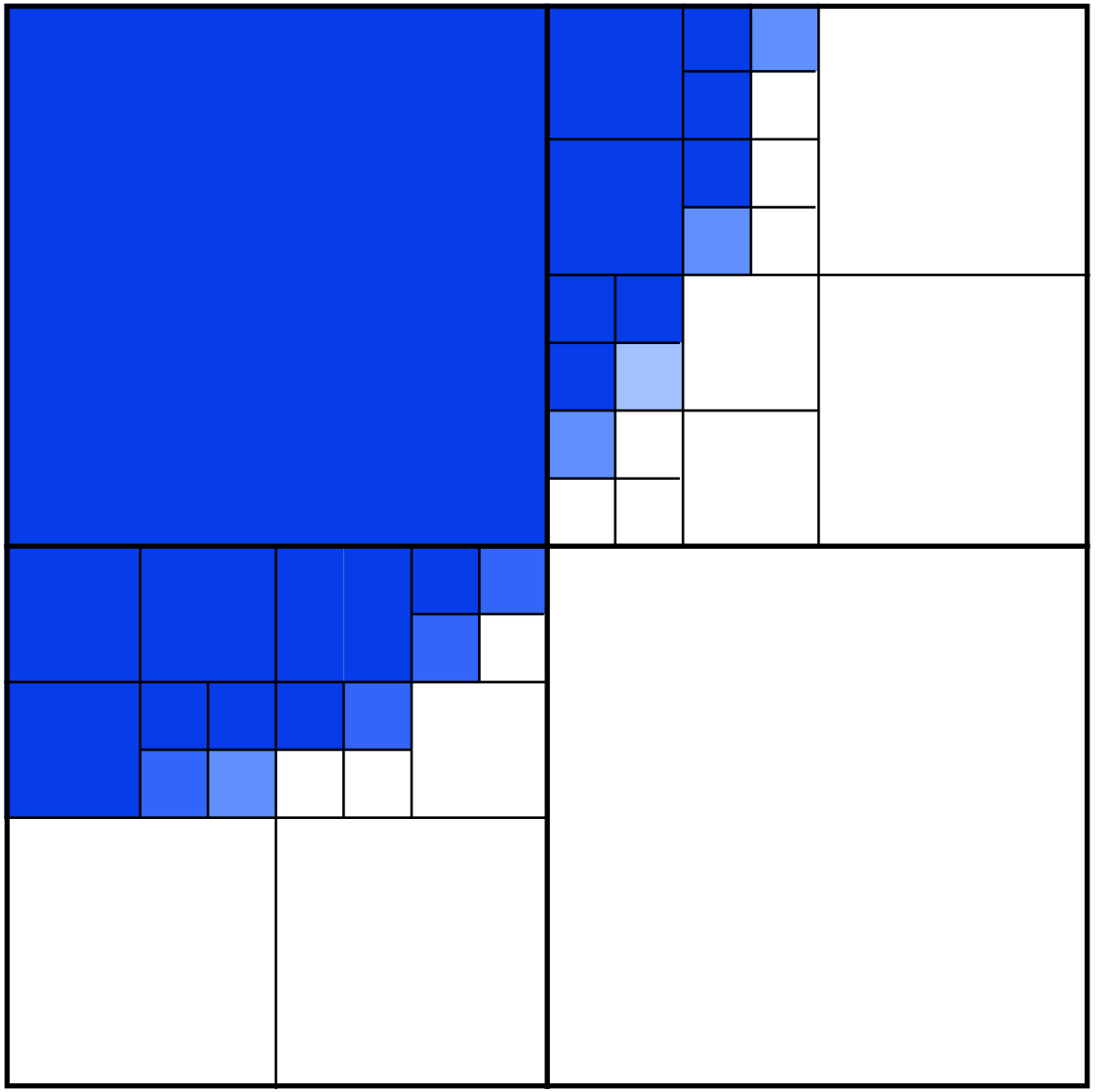
Pyramida

- **reprezentace plošných oblastí v rovině**
- **proti quadtree obsahuje navíc souhrnnou info. ve vnitřních uzlech**
- **uzly pyramidy až do úrovně E**



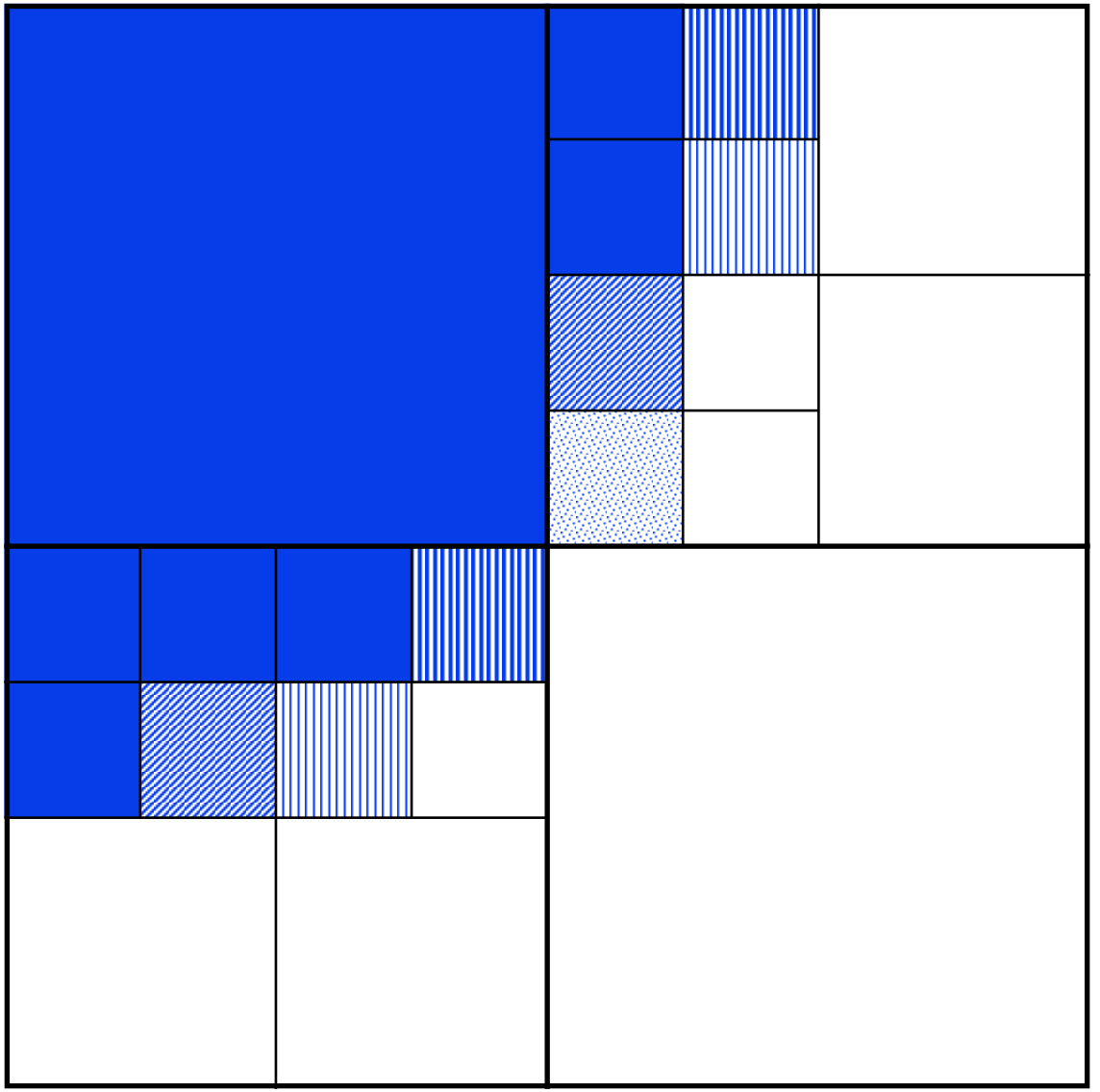
Pyramida

- **uzly pyramidy až do úrovně D**



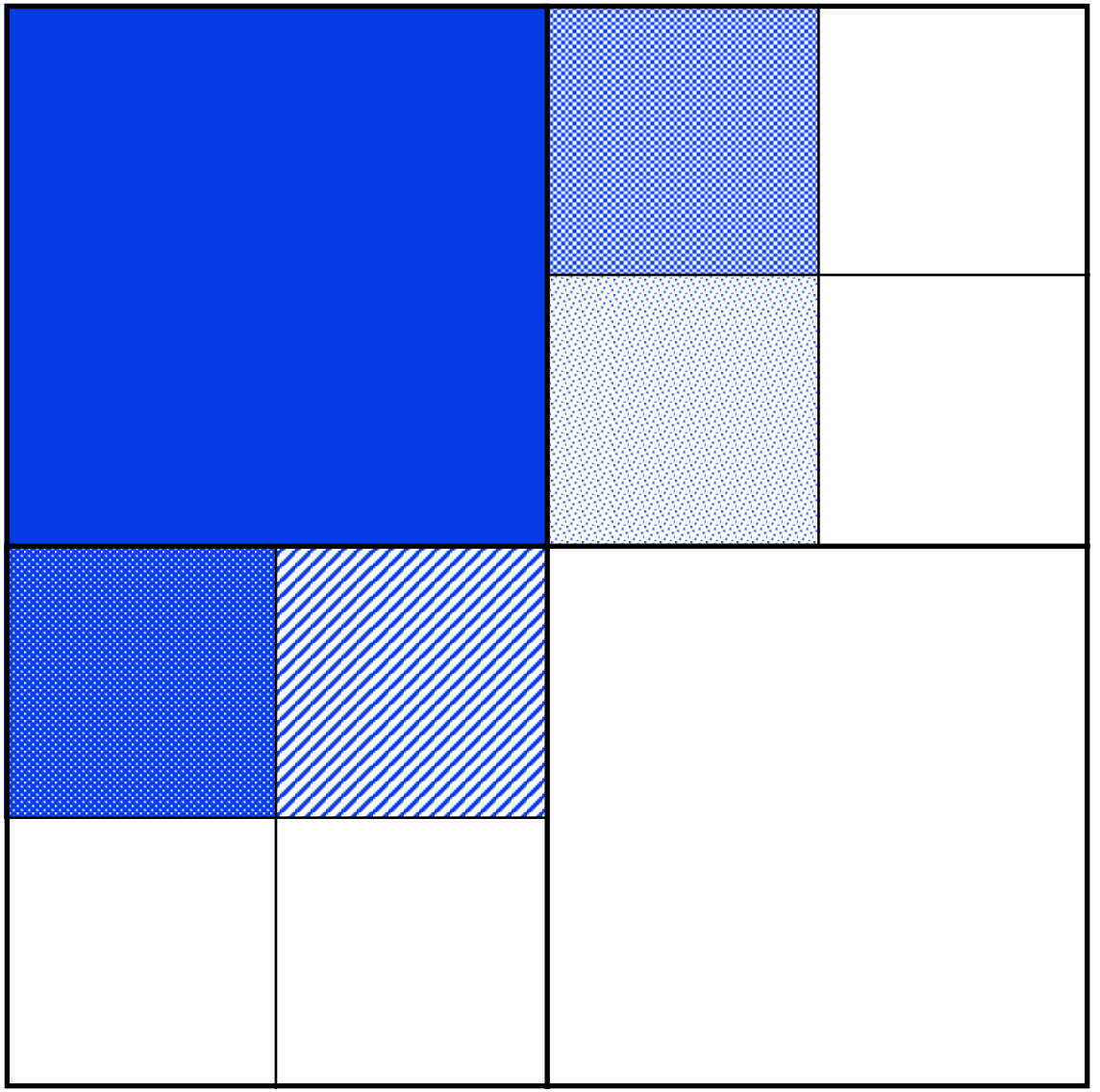
Pyramida

- **uzly pyramidy až do úrovně C**



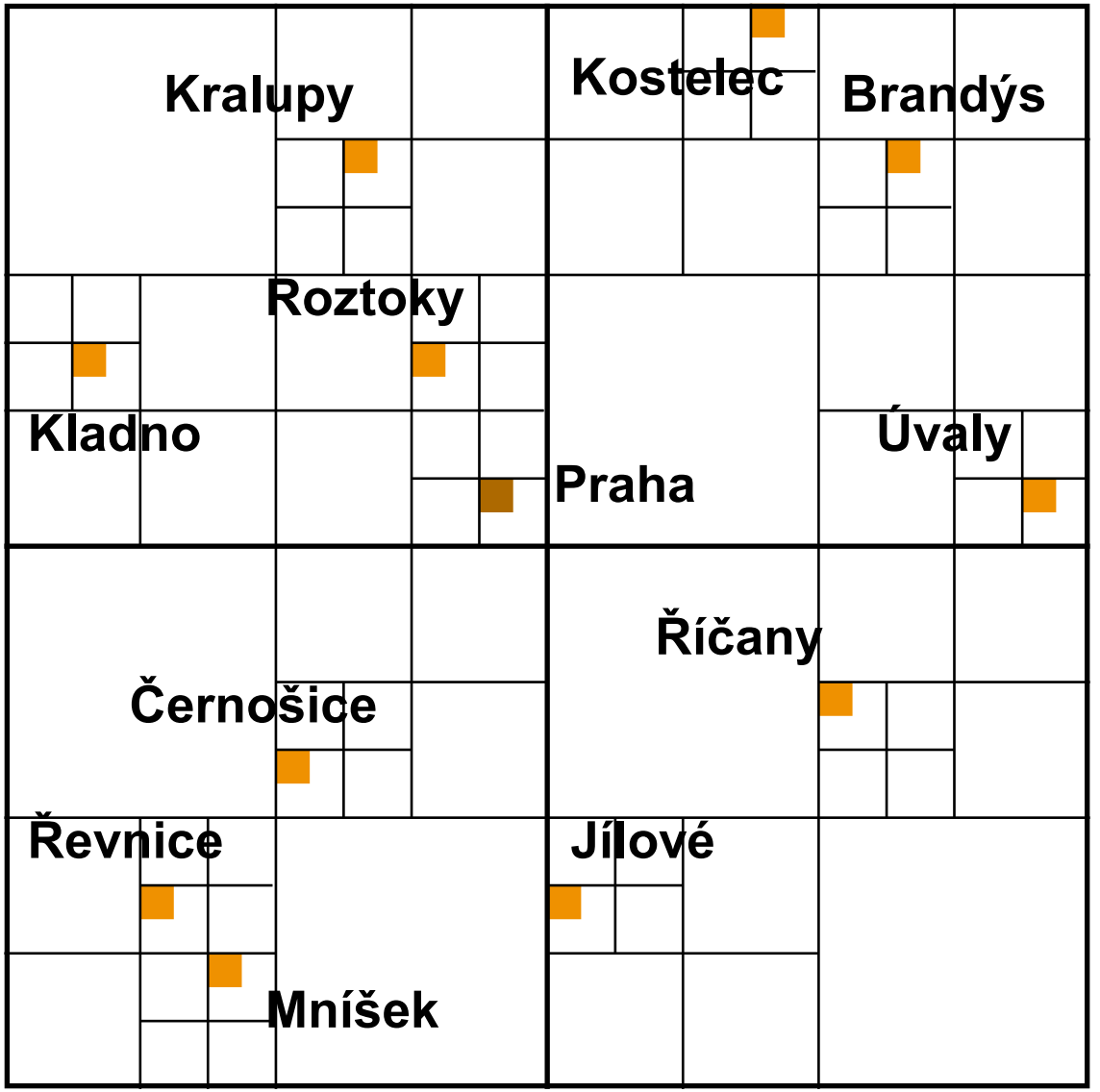
Pyramida

- **uzly pyramidy až do úrovně B**



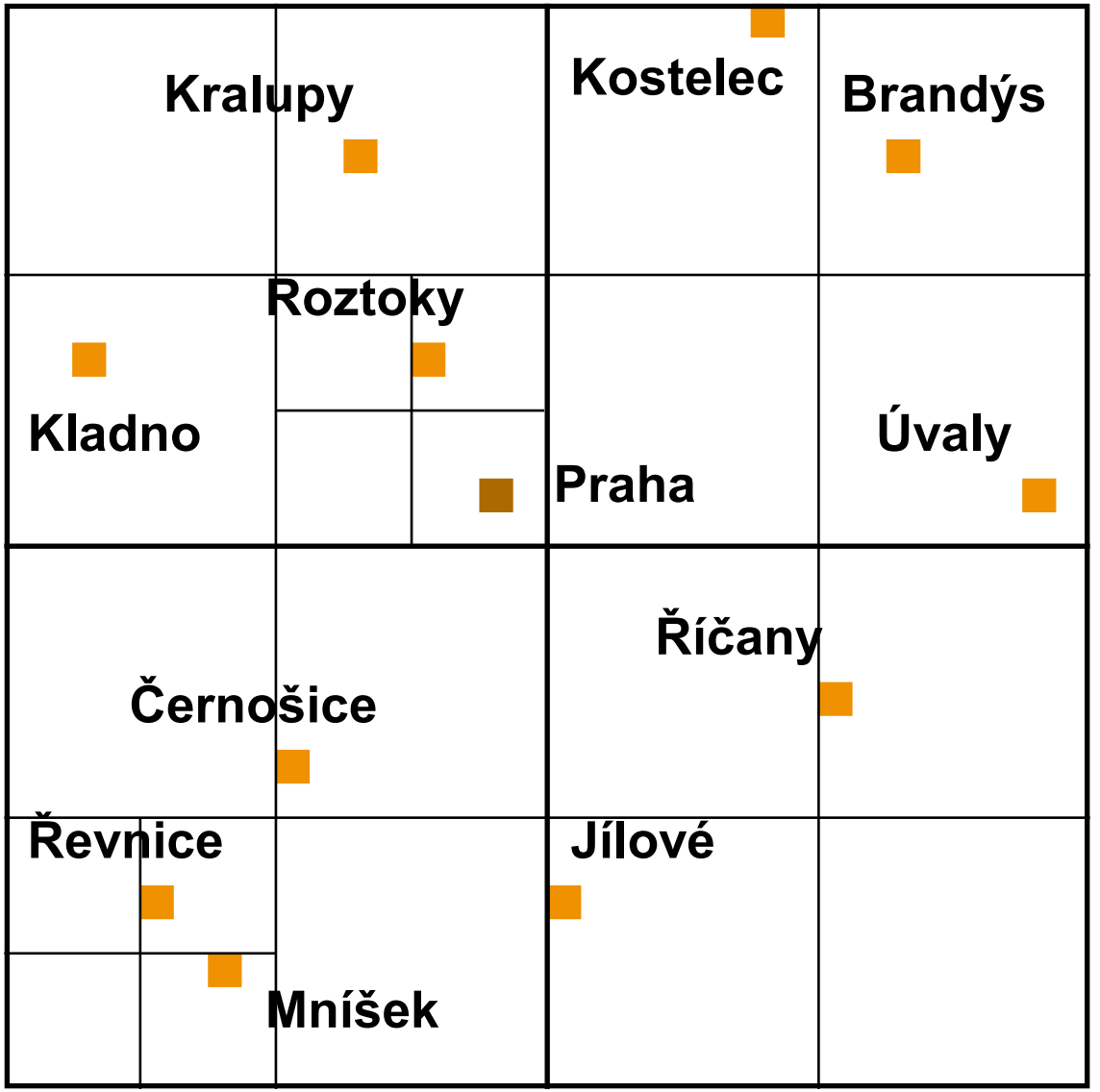
„MX quadtree” (matrix)

- reprezentace bodových objektů
- dělení se provádí přesně v polovině
- informace o objektech jsou uloženy v listech (stejné hloubky)

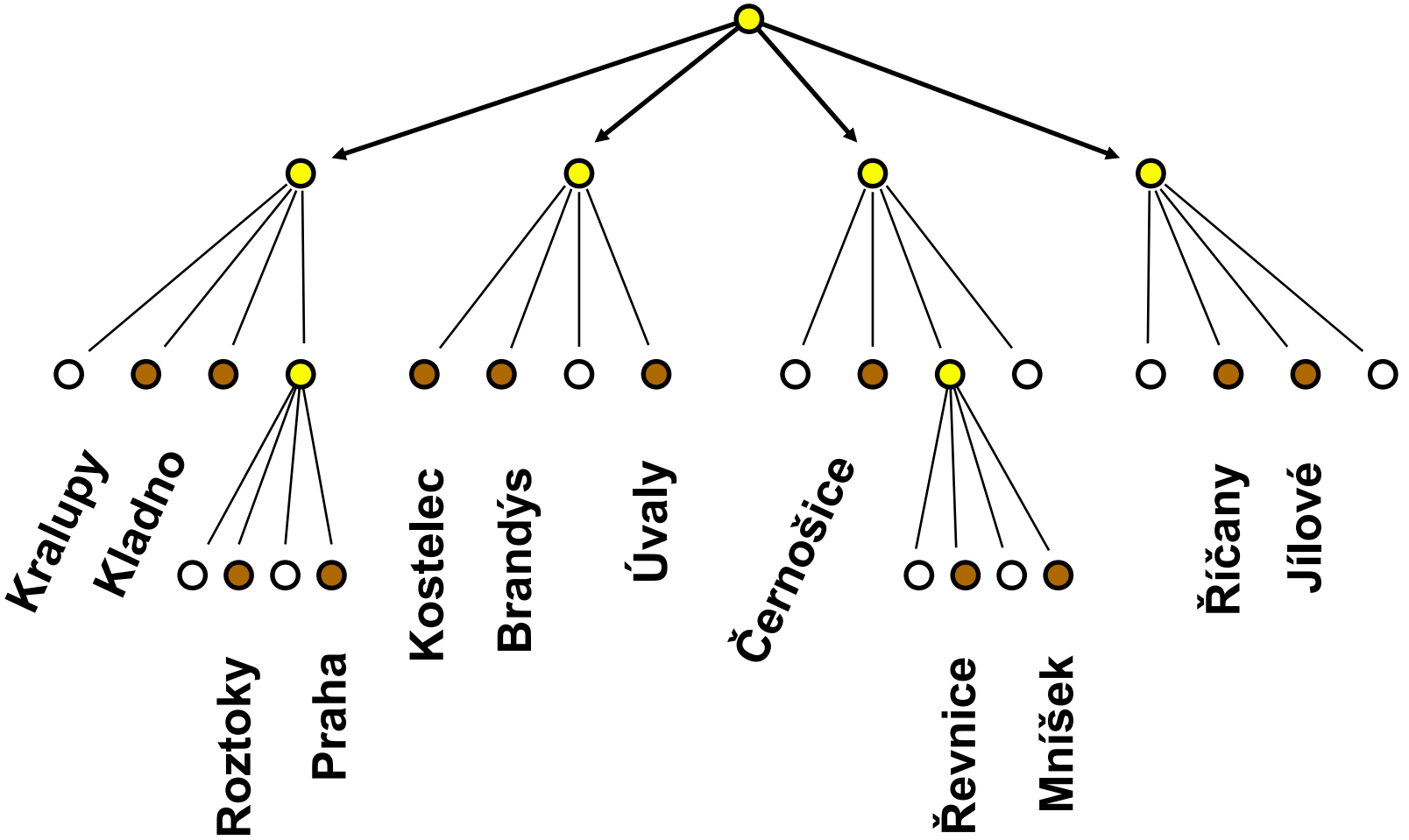


„PR quadtree” (point-region)

- reprezentace bodových objektů
- dělení se provádí přesně v polovině
- informace o objektech jsou uloženy v listech (všechna patra, max. 1 objekt/l.)



„PR quadtree” (Orenstein)



Metody s kbelíkem („bucket”)

- v jednom listu (uzlu) se ukládají **údaje o několika objektech**

  0 < počet objektů ≤ Max

  – konstanta **Max** se volí s ohledem na fyzickou implementaci (velikost záznamu)

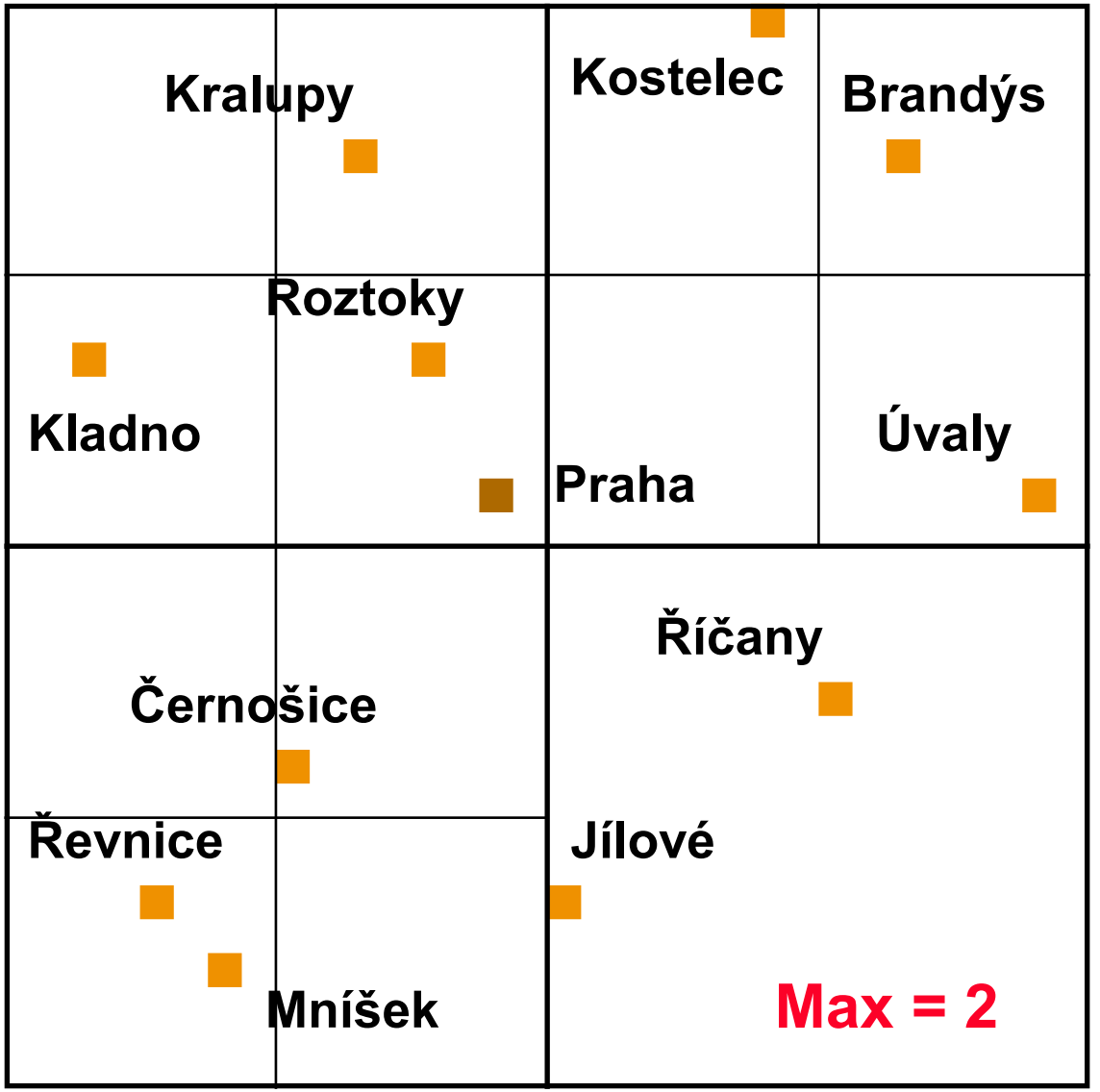
- zmenšení spotřeby **paměti i času přístupu**

  – menší režie na údržbu ukazatelů

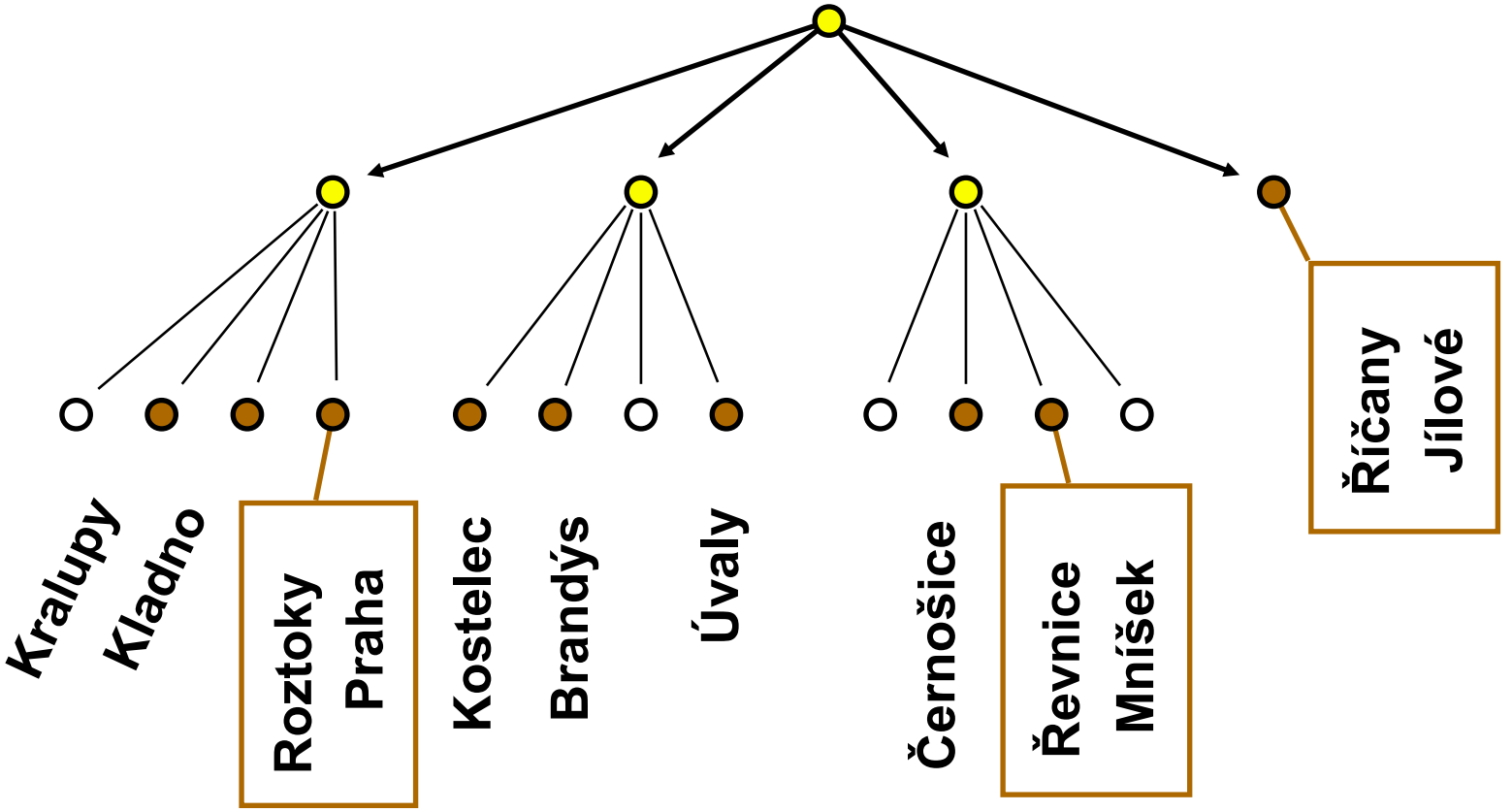
- analogie **A-B stromů** (jako alternativy běžných binárních vyhledávacích stromů)

„Bucket PR quadtree”

- reprezentace bodových objektů
- dělení se provádí přesně v polovině
- informace o objektech jsou uloženy v listech (všechna patra, Max objektů/l.)



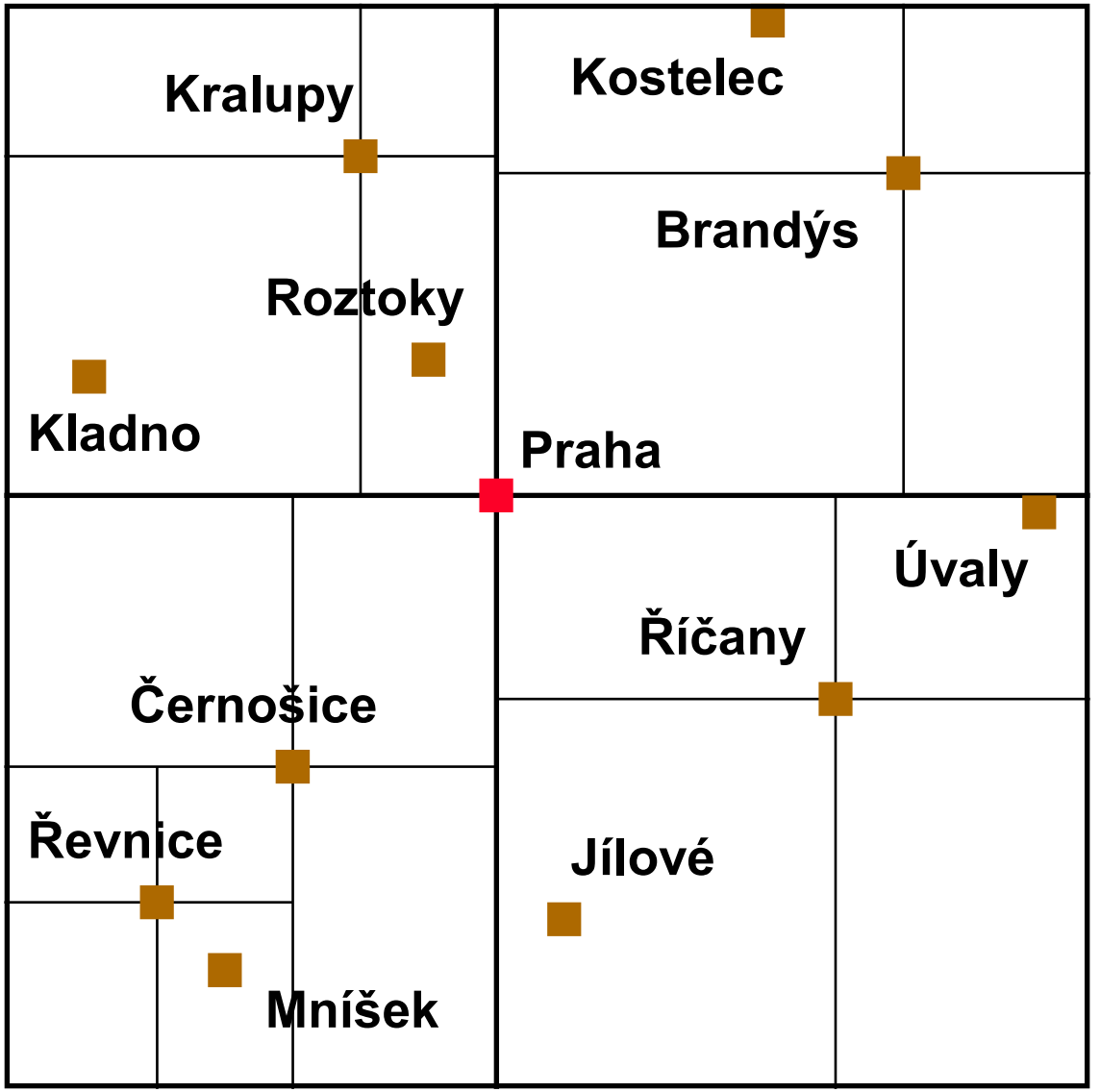
„Bucket PR quadtree”



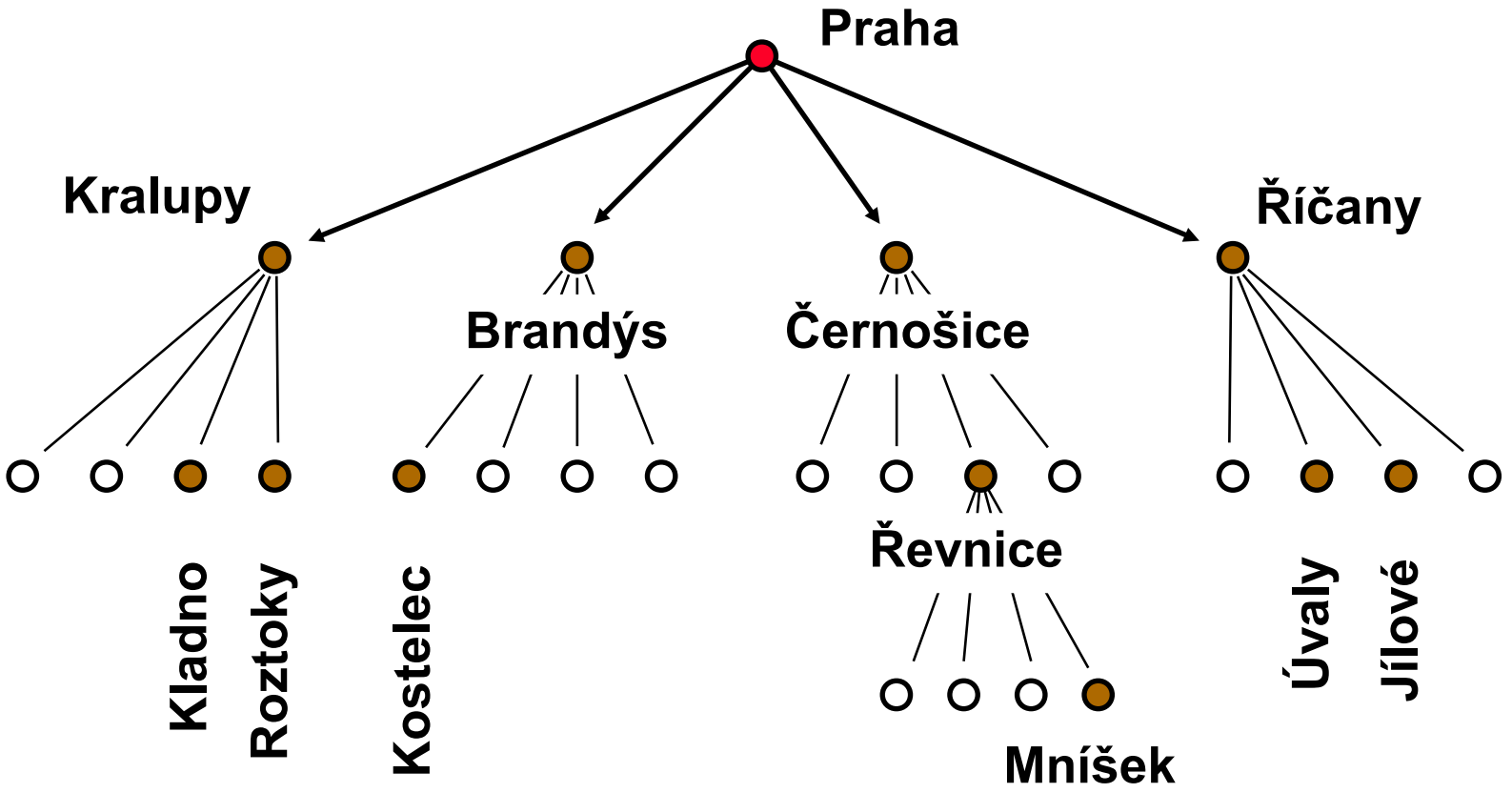
Max = 2

„Point quadtree” (Finkel/Bentley)

- **reprezentace bodových objektů**
- **dělení se adaptuje podle polohy objektů**
- **informace o objektech se ukládají i do vnitřních uzlů**

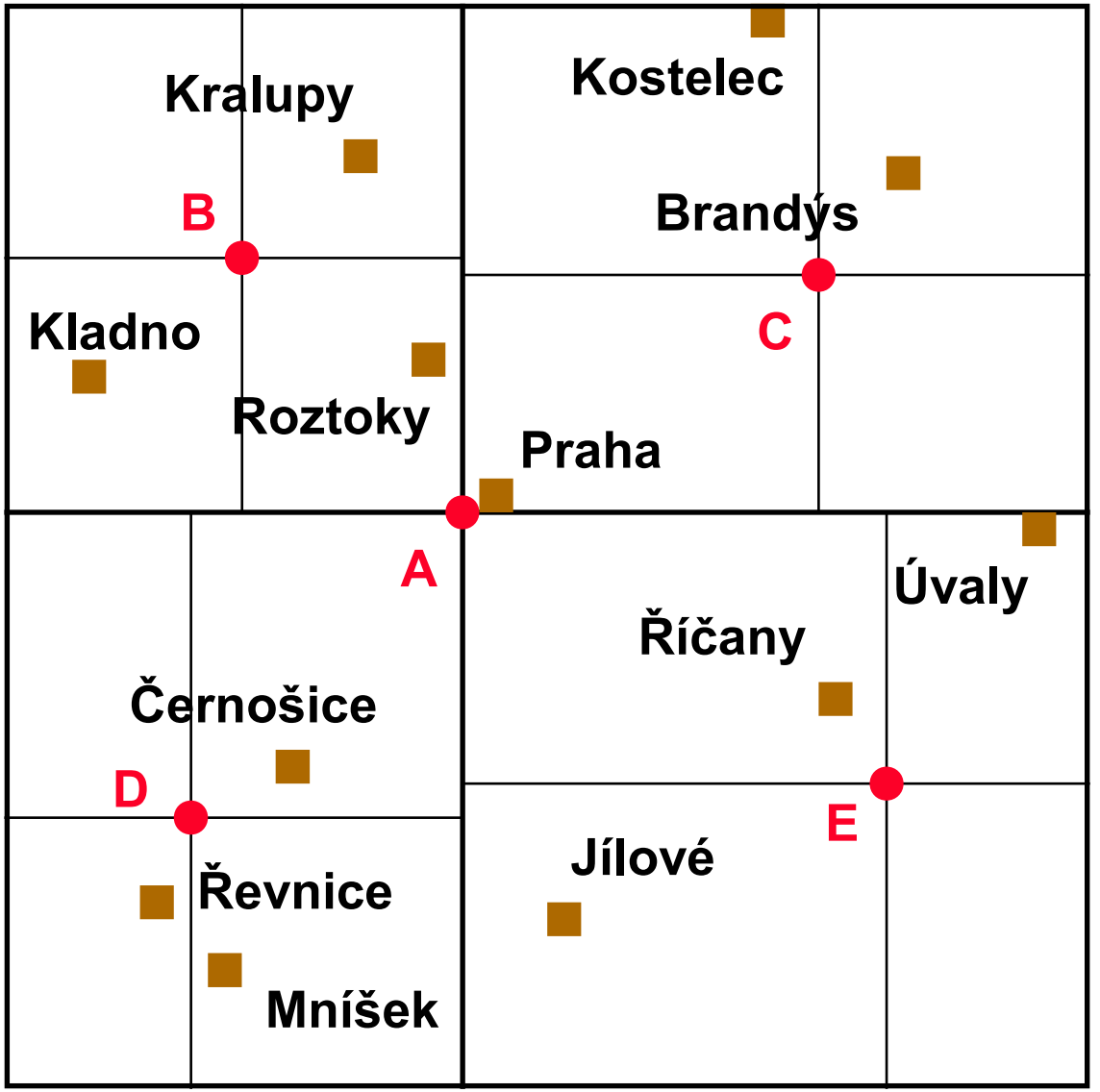


„Point quadtree”

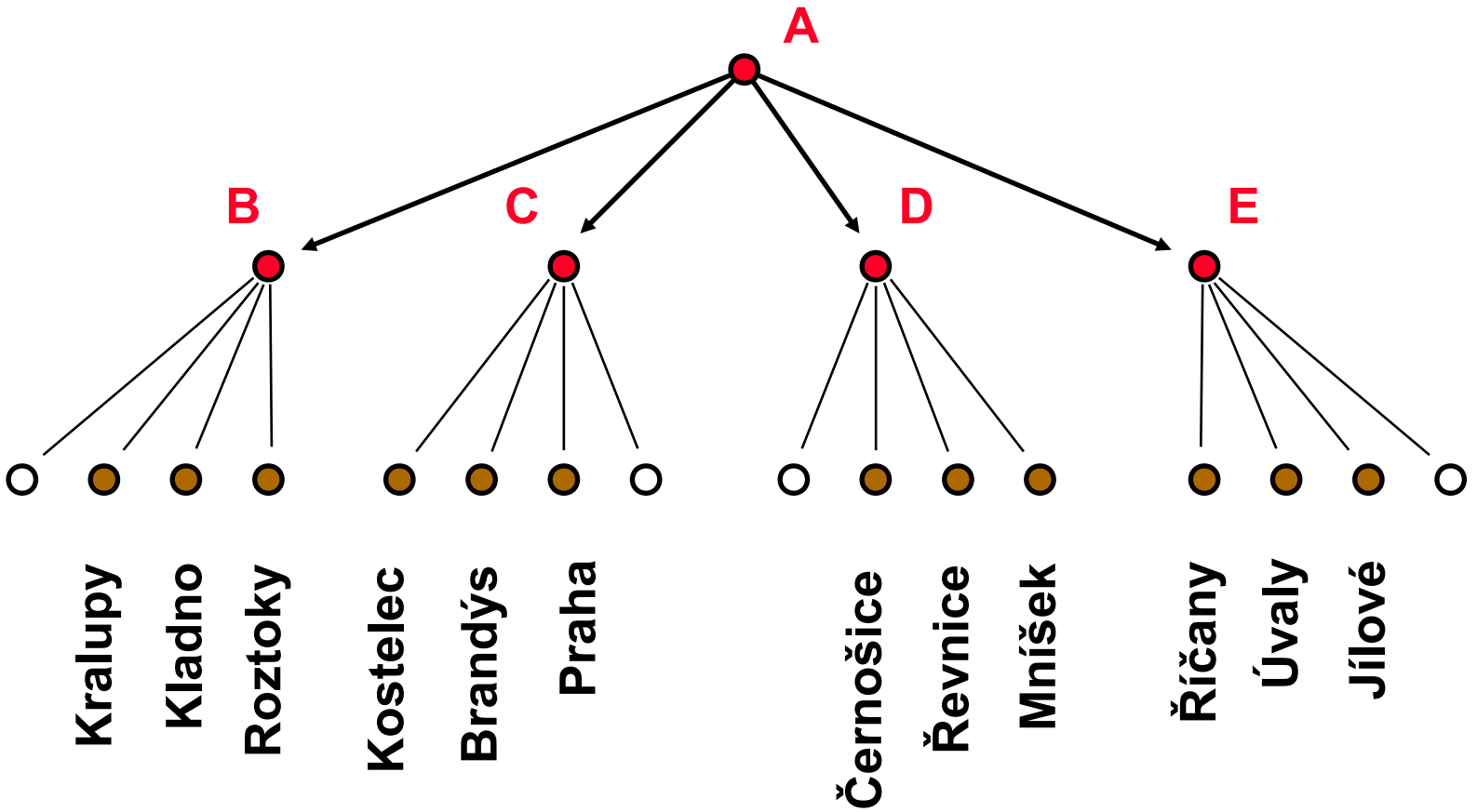


„Pseudo quadtree” (Overmars)

- reprezentace <u>bodových objektů</u>
- dělení se adaptuje podle <u>polohy objektů</u> (dělí se <u>mimo objekty</u>)
- objekty se ukládají pouze <u>do listů</u>

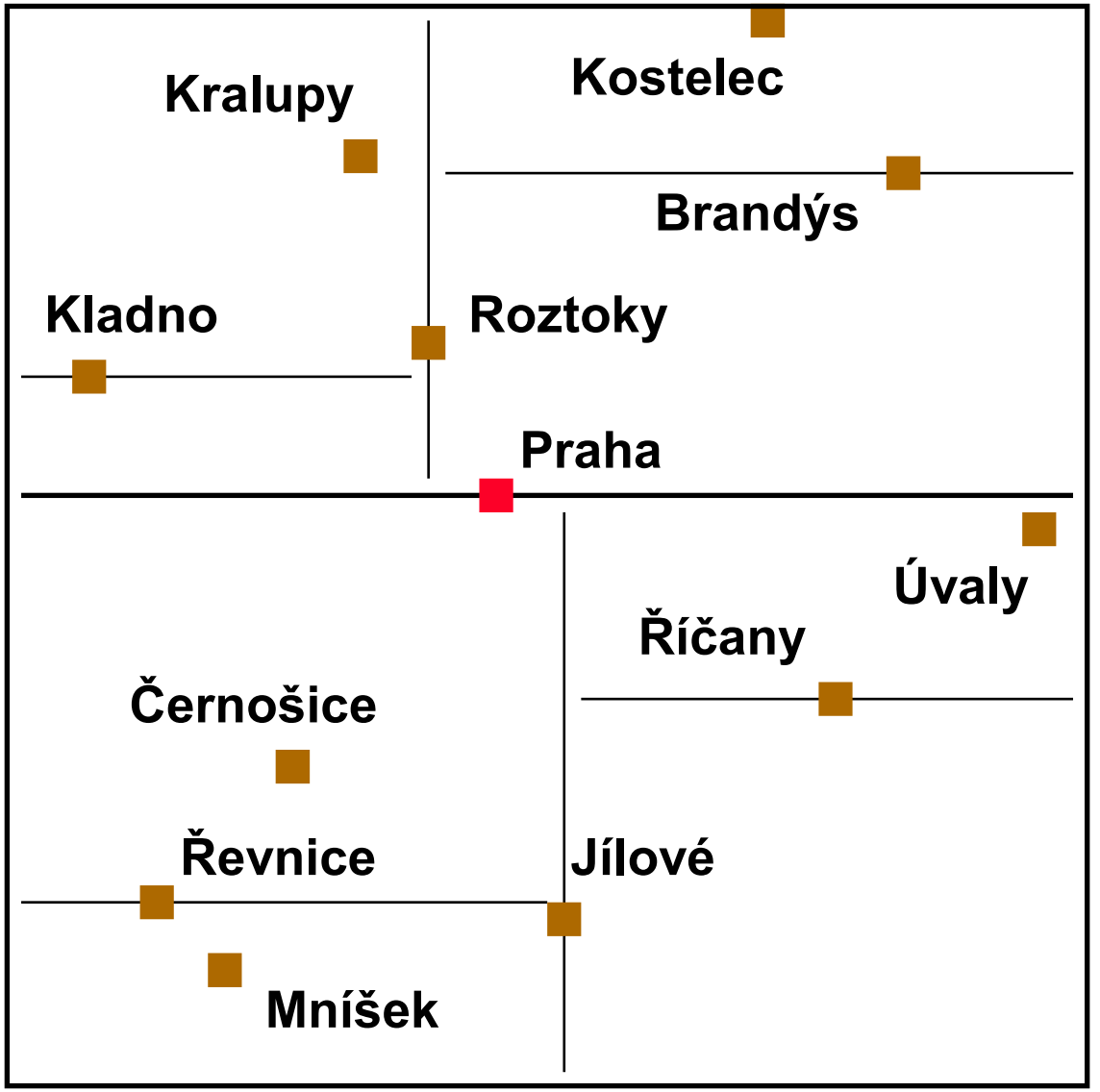


„Pseudo quadtree”

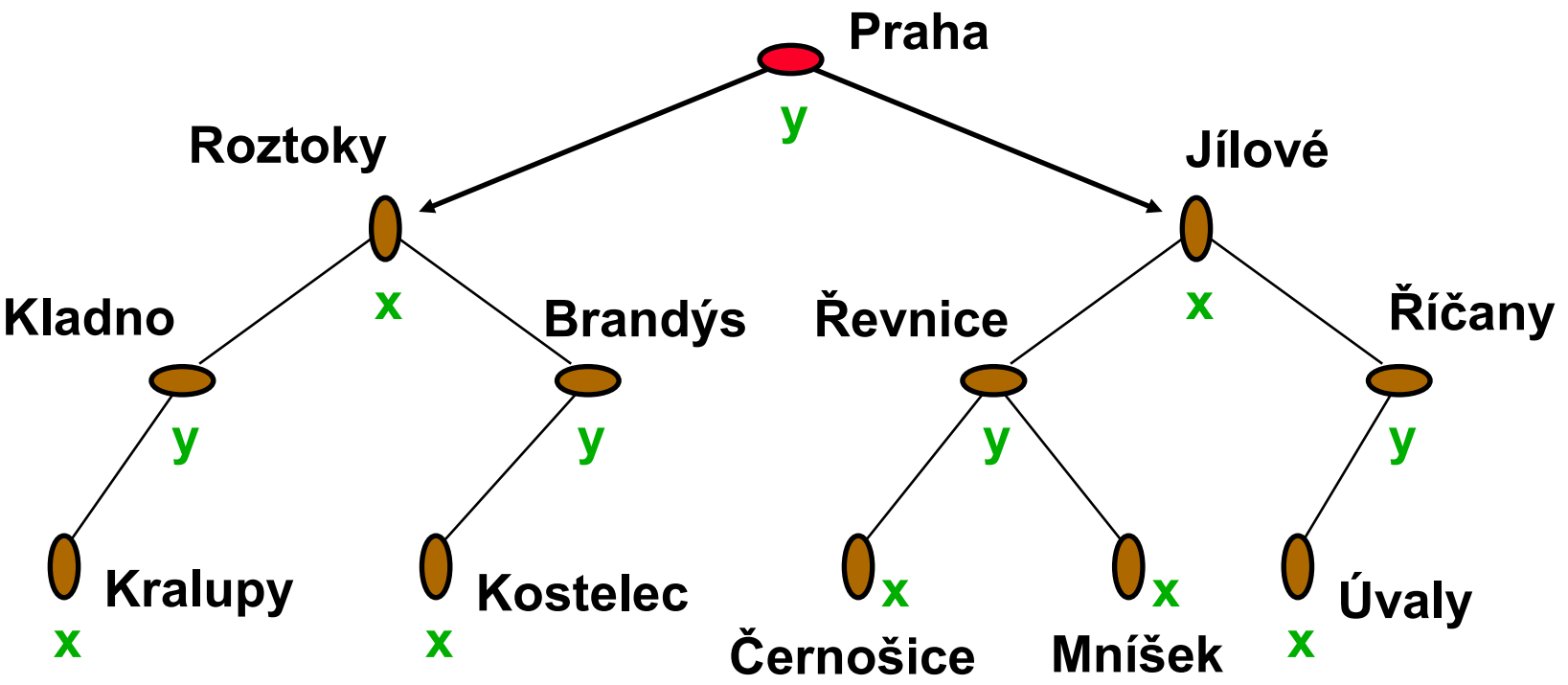


„K-D tree” (Bentley)

- reprezentace bodových objektů
- adaptivní dělení pouze podle jedné složky (binární strom), pravidelné střídání souř. složek
- objekty jsou ve všech uzlech

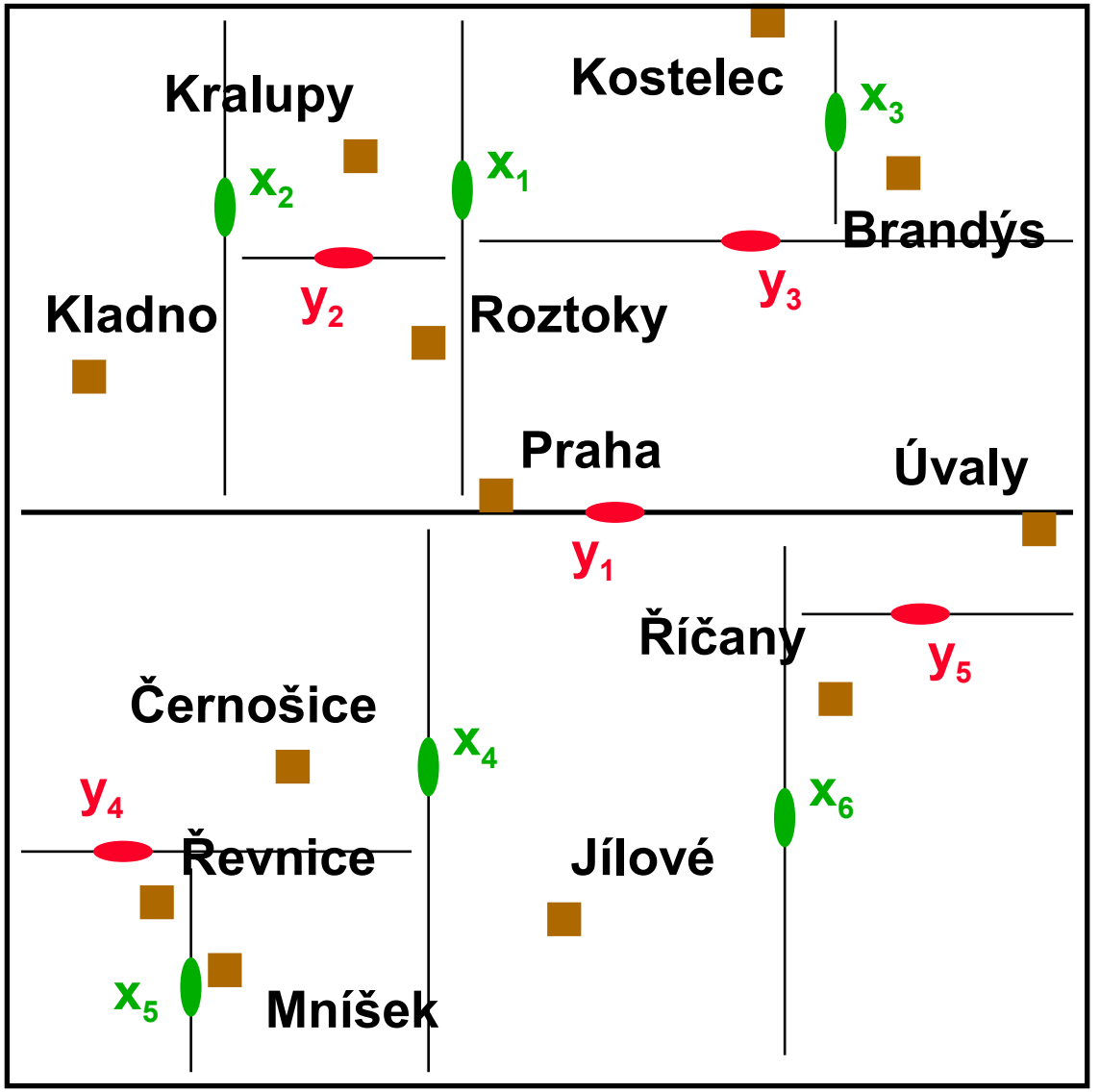


„K-D tree”

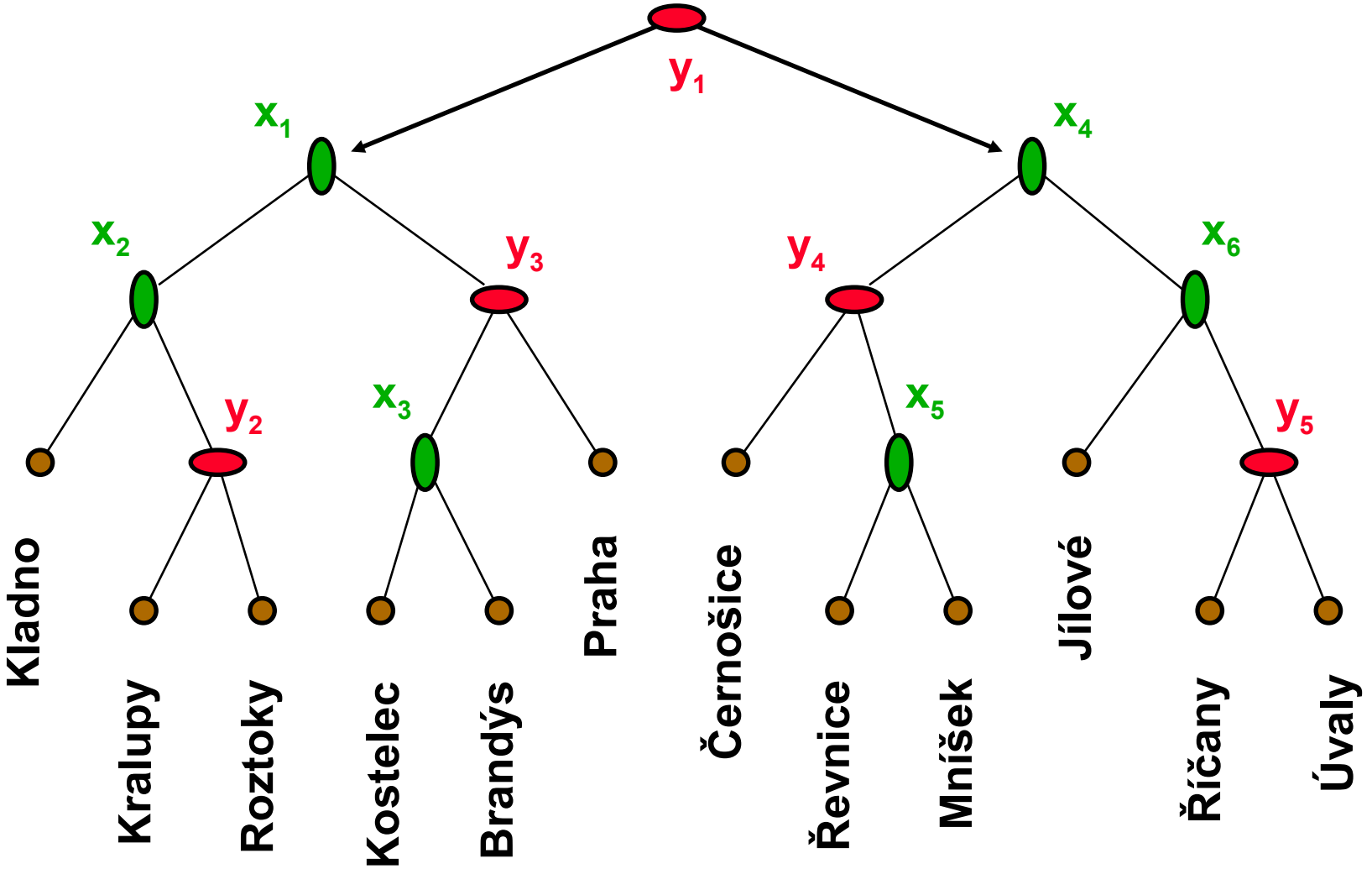

Praha
y
Roztoky
Jílové
x
x
Kladno
Brandýs
Řevnice
Říčany
y
y
y
y
Kralupy
Kostelec
Černošice
Mníšek
Úvaly
x
x
x
x
x

„Adaptive K-D tree” (Friedman)

- reprezentace bodových objektů
- adaptivní dělení pouze podle jedné složky (binární strom) mimo objekty (medián)
- objekty jsou pouze v listech

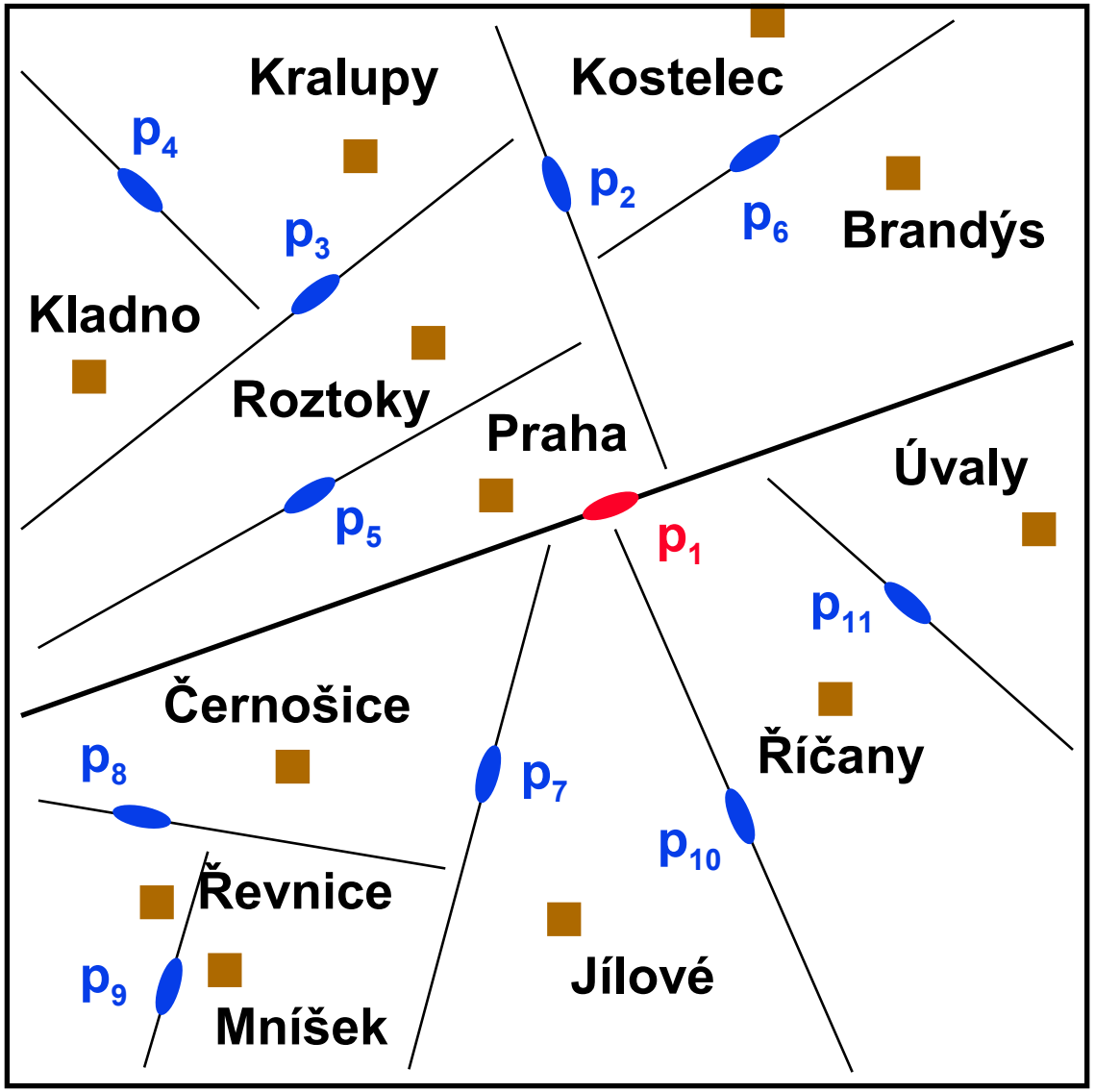


„Adaptive K-D tree”

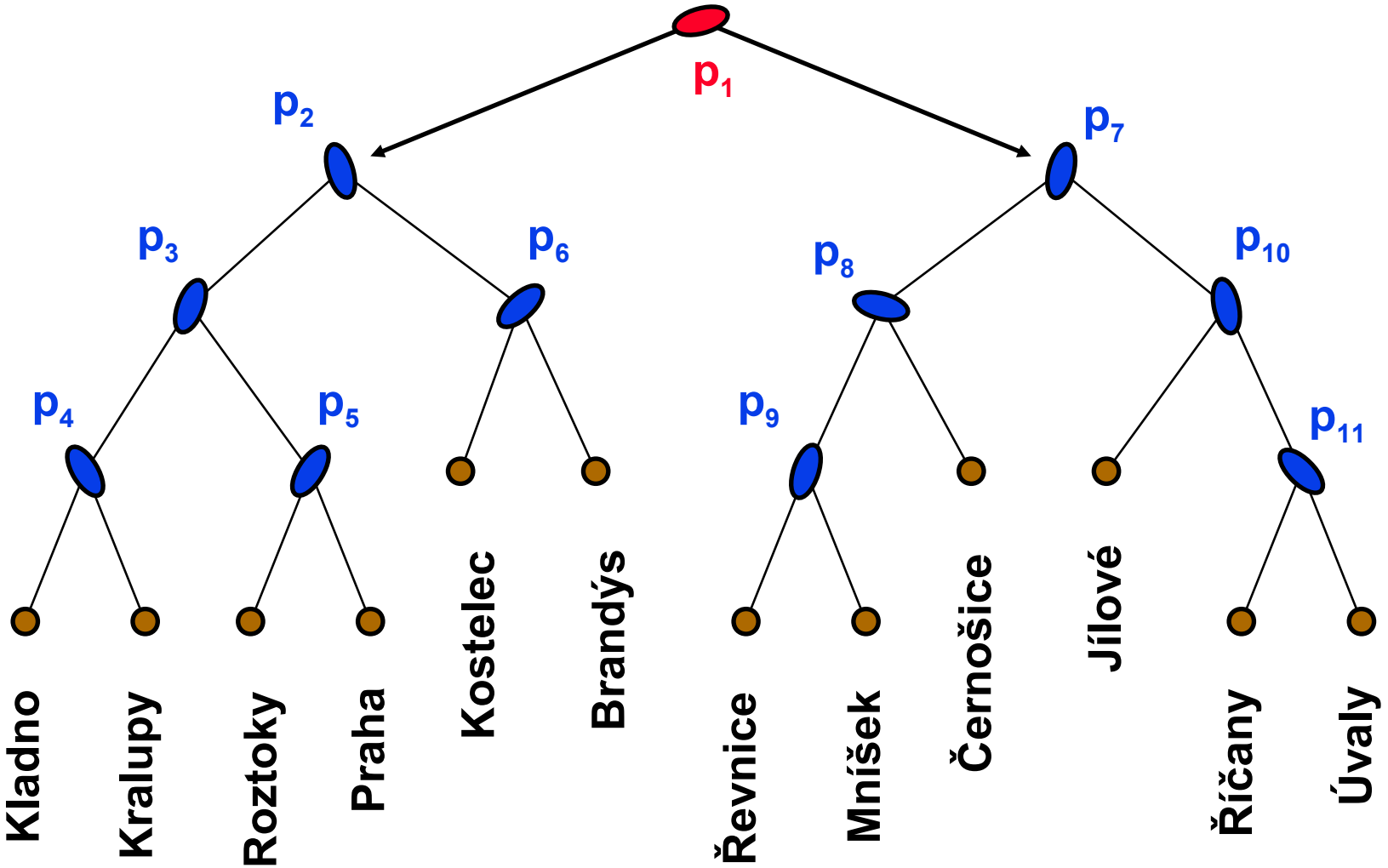


„BSP tree” (Fuchs, Kedem, Naylor)

- **rozklad prostoru na <u>konvexní oblasti</u>**
- **adaptivní dělení <u>libovolnou nad-rovinou</u>**
- **bodové objekty jsou uloženy <u>v listech</u>**



„Point BSP tree”

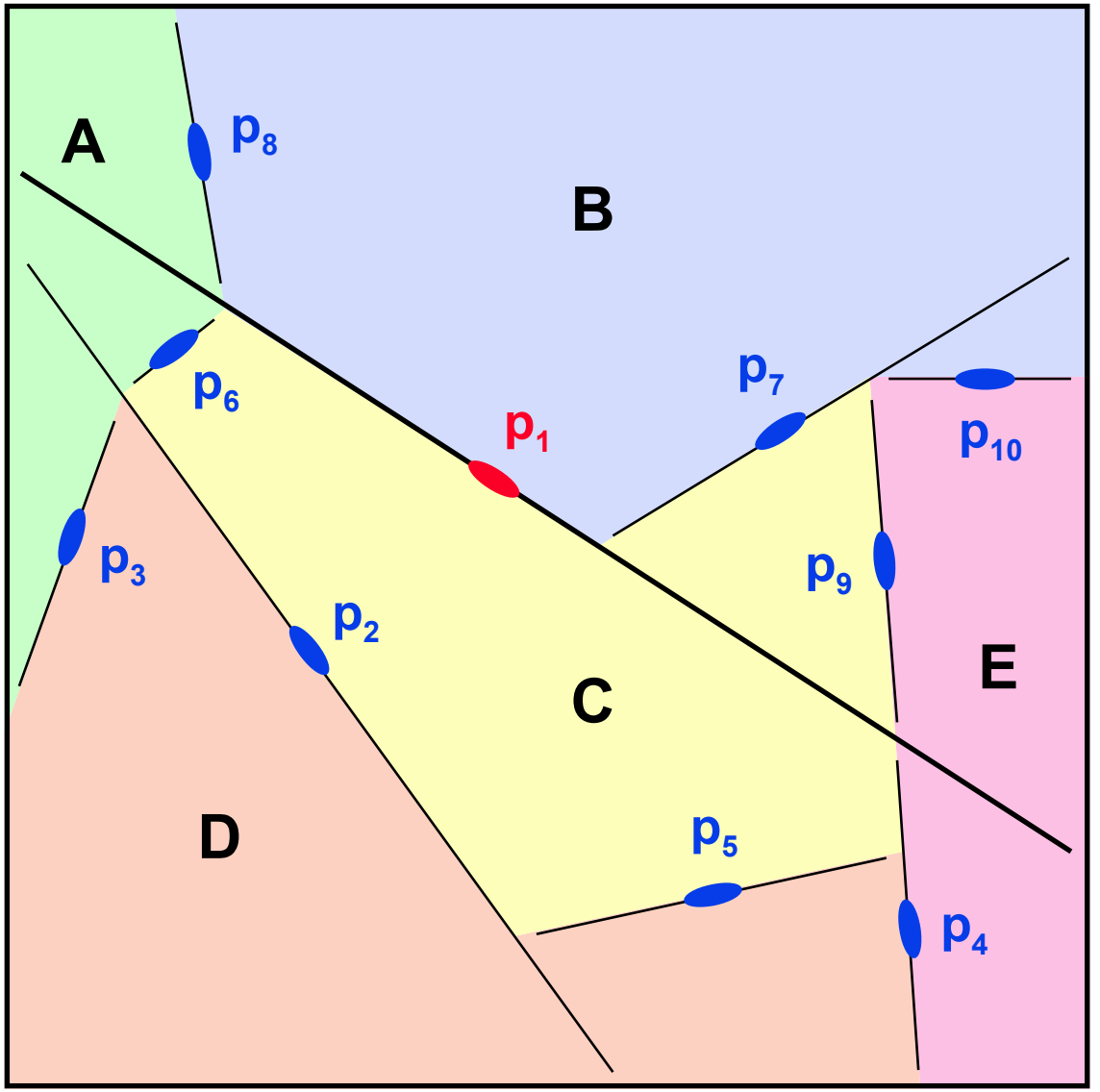


„Region BSP tree”

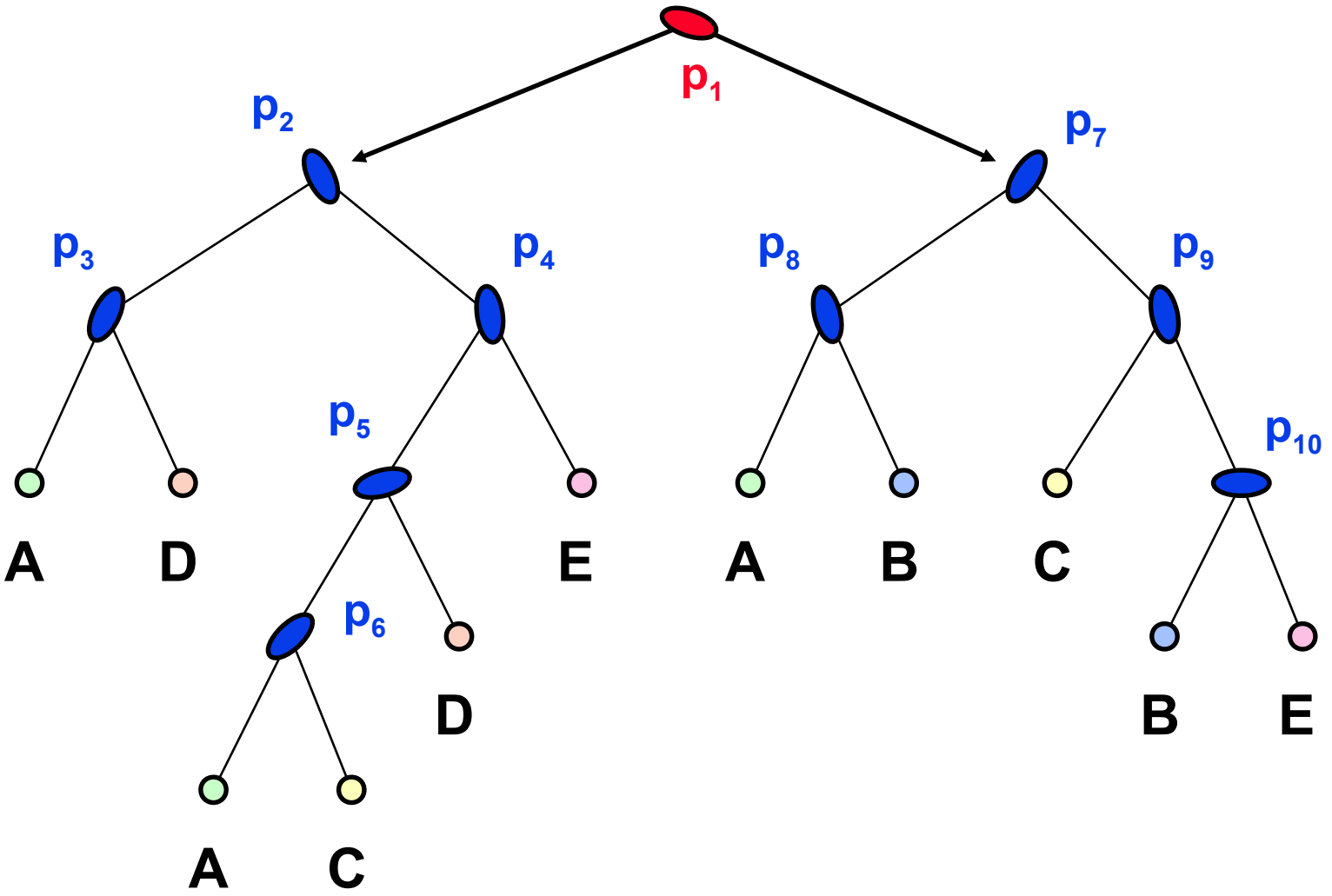
- slouží k reprezentaci **polygonálního rozkladu** 2/3D prostoru
  - jednotlivé buňky (polygony/polyhedry) nemusí být konvexní
- reprezentace **polygonální 2/3D scény**
  - ve vnitřních uzlech jsou navíc uloženy informace o hraničních hranách/stěnách
- **množinové operace** nad polygony/polyhedry
  - převod CSG → B-rep, výpočet vržených stínů (stínová tělesa)

„Region BSP tree”

- **reprezentace <u>polygonální sítě</u>**
- **dělicí nadrovina prochází nějakou <u>hranicí dělení</u> (uzel může obs. popis hrany)**
- **listy obsahují <u>identifikaci polygonu</u>**



„Region BSP tree”

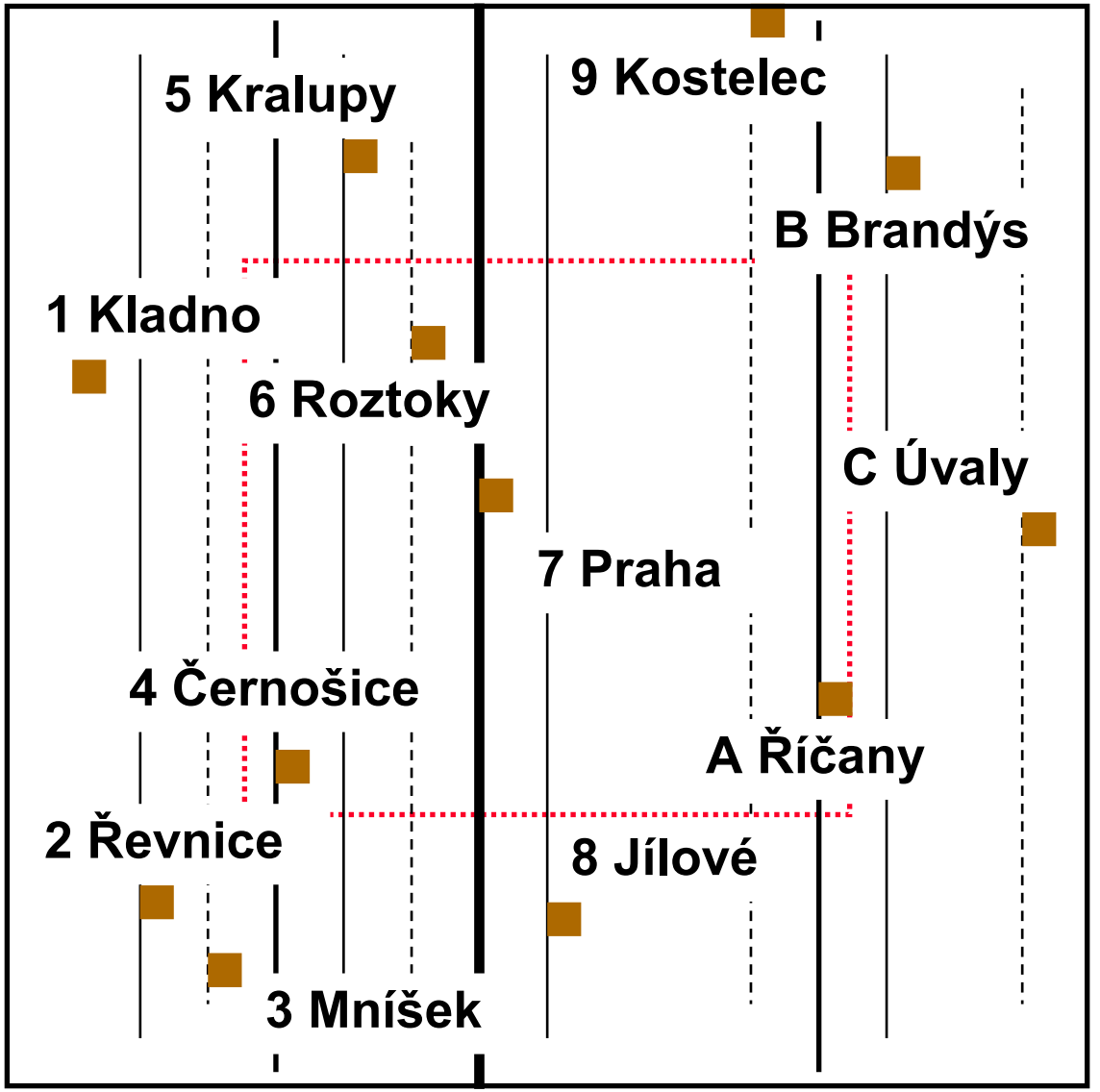


„Range trees”

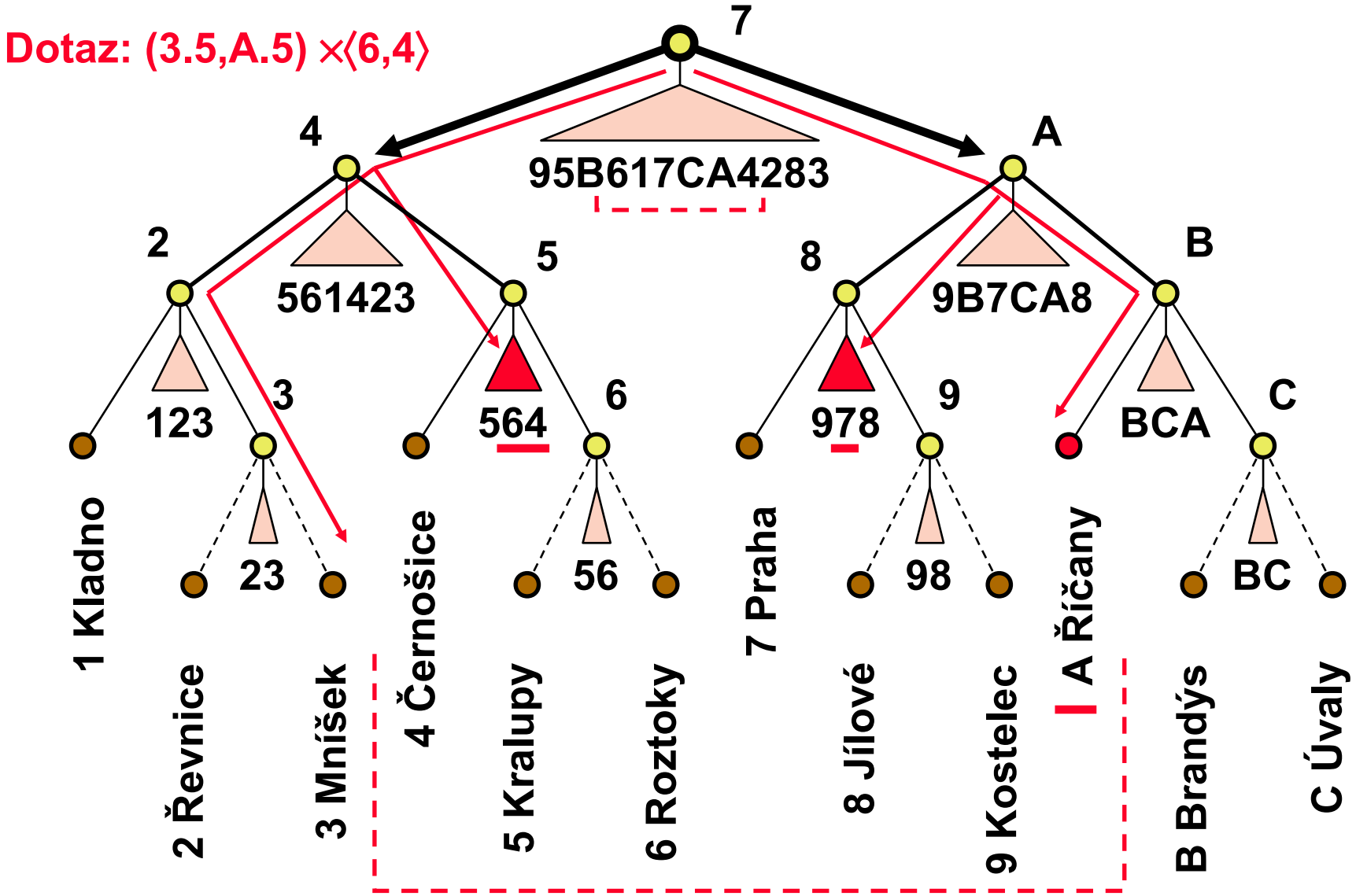
- rychlé implementace **intervalových dotazů**
  - $\langle x_{min}, x_{max} \rangle \times \langle y_{min}, y_{max} \rangle$ ve 2D
  - složitost $O(\log_2 N + F)$ v 1D, $O(\log_2^2 N + F)$ ve 2D
- **„1D range tree”**
  - vyvážený binární vyhledávací strom s listy propojenými obousměrným spojovým seznamem
- **„2D range tree”**
  - vyvážený binární vyhledávací strom pro souřadnici **x**
  - každý vnitřní uzel obsahuje „1D range tree” (souř. **y**) bodů obsažených v příslušném podstromu

„2D range tree”

- **vyvážený binární vyhledávací strom podle souřadnice x**
- **objekty jsou pouze v listech**
- **vnitřní uzly obsahují 1D „range trees” podle souř. y**

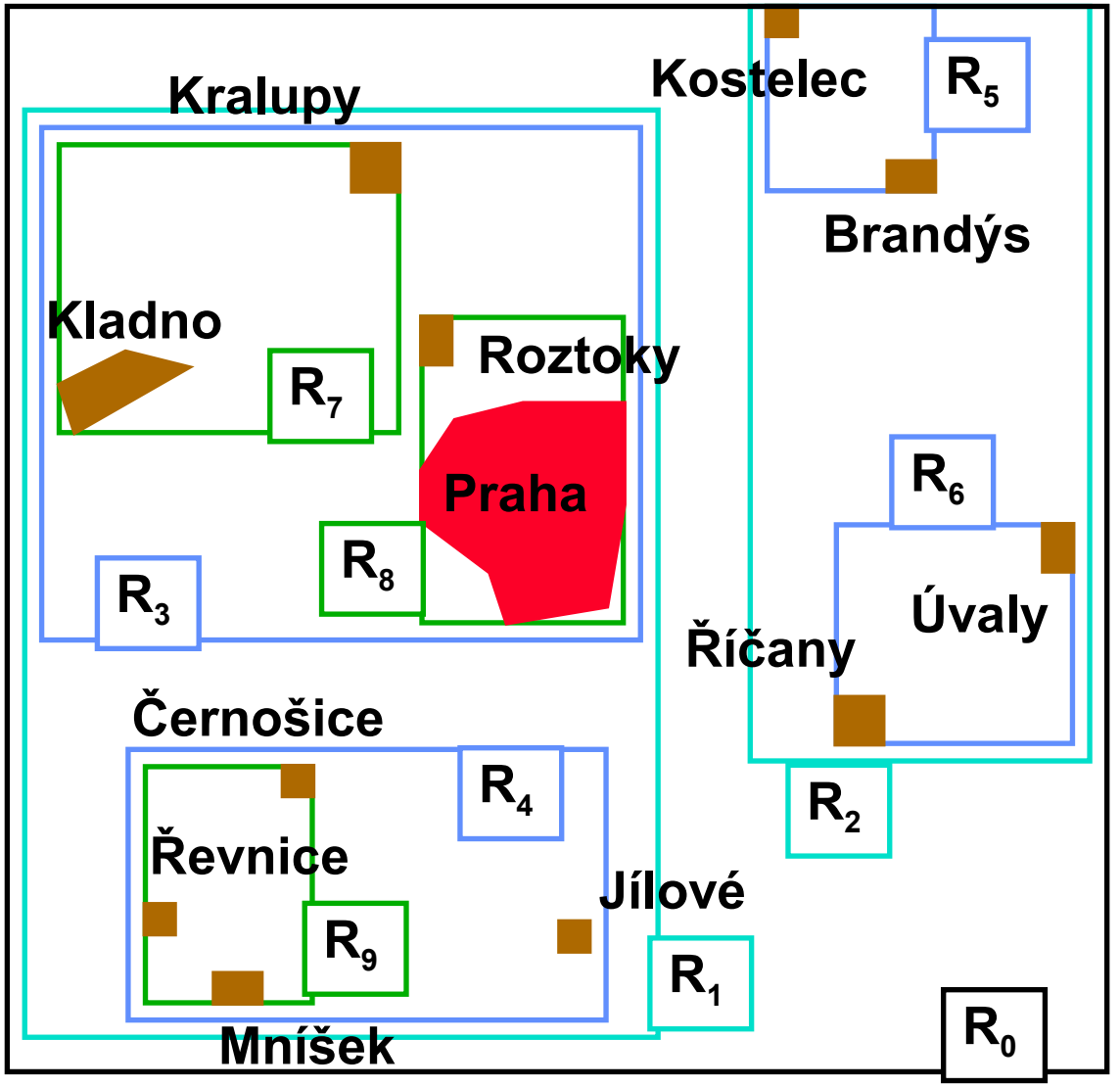


„2D range tree”

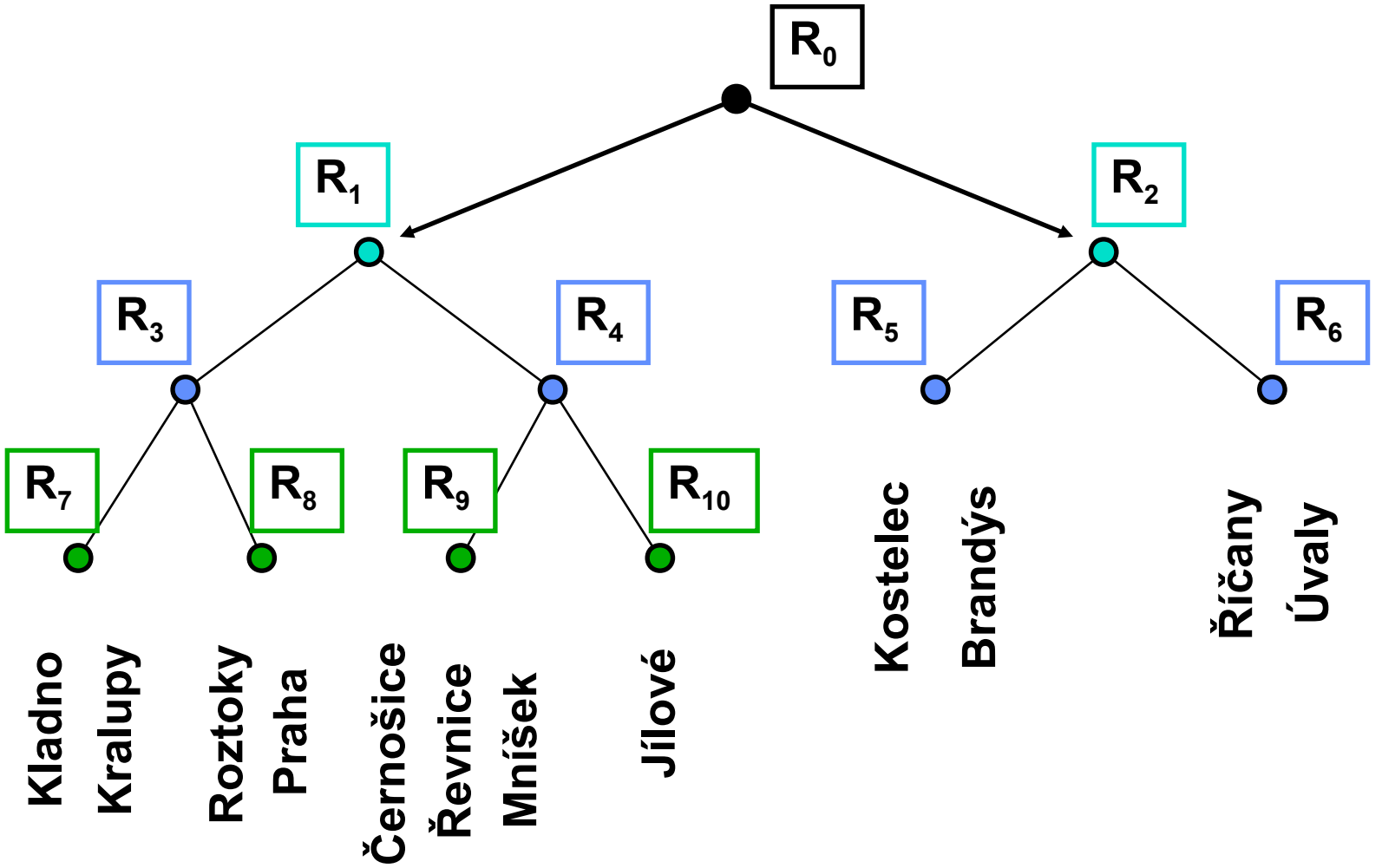


„R-tree” (Guttman)

- reprezentace plošných objektů
- vnitřní uzel obsahuje obalový obdélník pod-stromu (organiz. jako B-strom)
- ukazatele na objekty jsou v listech

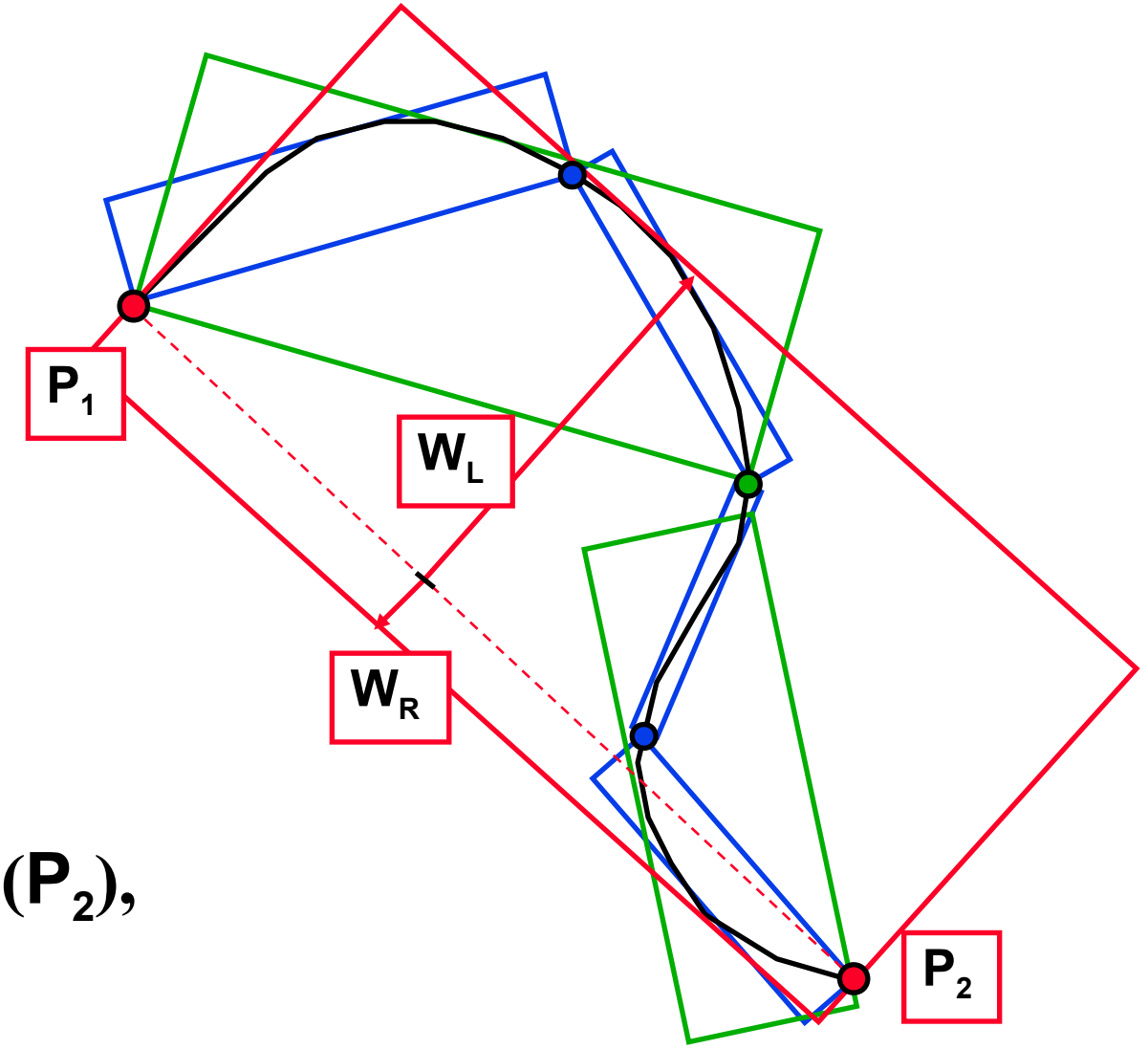


„R-tree”


R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
Kladno
Kralupy
Roztoky
Praha
Černošice
Řevnice
Mníšek
Jílové
Kostelec
Brandýs
Říčany
Úvaly

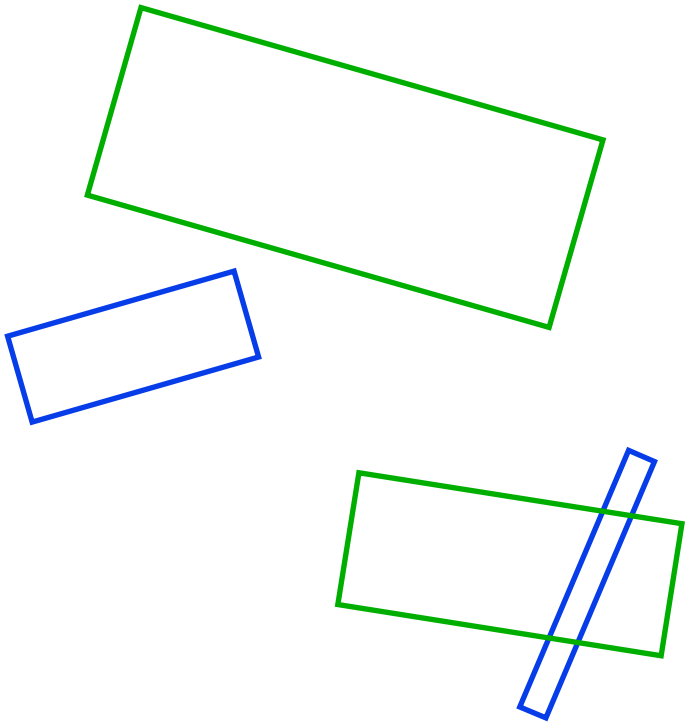
„Strip tree" (Ballard)

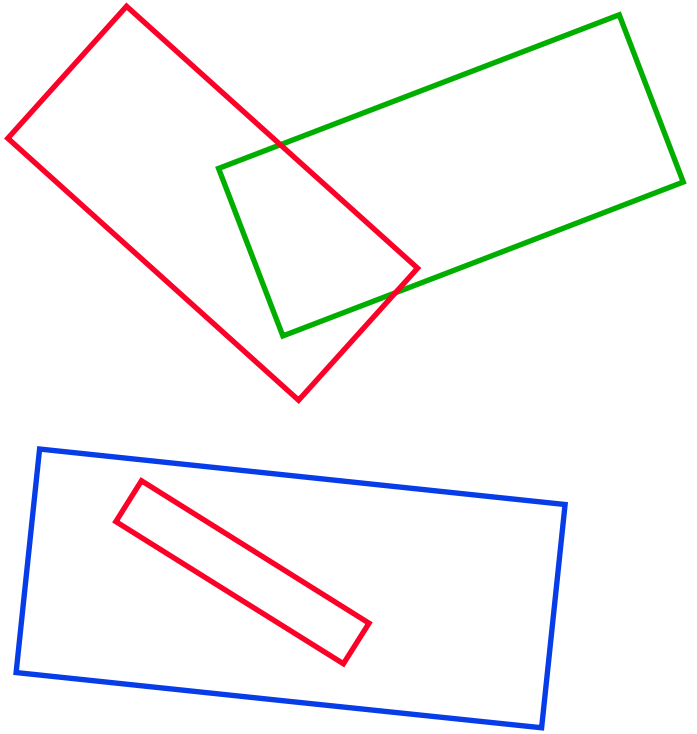
- **reprezentace <u>křivky</u> (lomené čáry) v rovině**
- **možnost adaptivního dělení křivky podle její <u>křivosti</u>**
- **pro každý obalový obdélník se ukládá: <u>začátek</u> (P_1), <u>konec</u> (P_2), <u>šířky</u> (W_L, W_R)**



Test průsečíku dvou křivek

- obě křivky mají postaven „**strip tree**”
  - **triviální případy**
  - nutno **dále dělit**



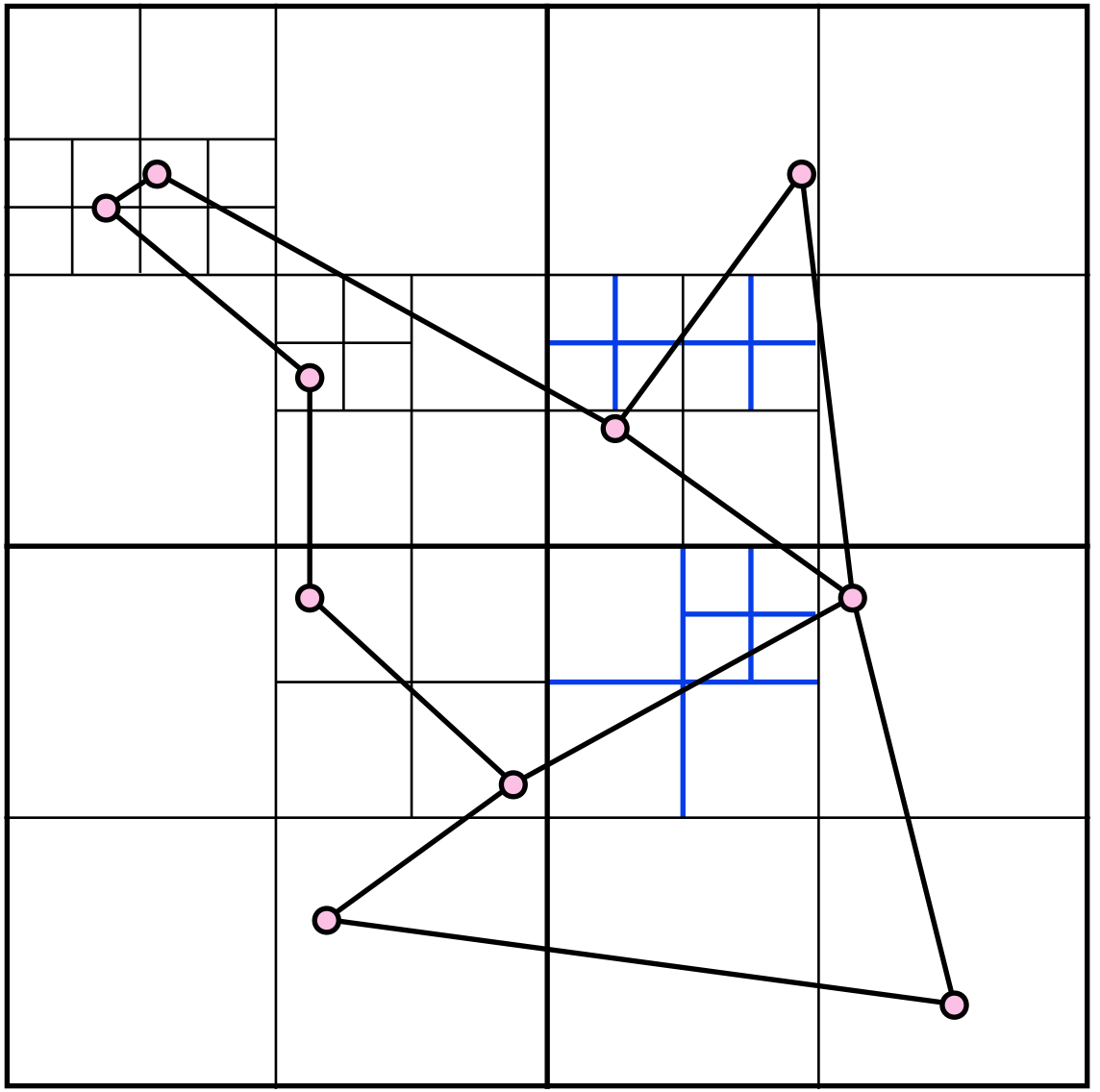


Hierarchické obalové systémy

- „**Sphere tree**” (Palmer, Grimsdale, 1995)
  - jednoduchý test i transformace, horší aproximace
- „**AABB tree**” (Held, Klosowski, Mitchell, 1995)
  - jednoduchý test, složitější transformace
- „**OBB tree**” (Gottschalk, Lin, Manocha, 1996)
  - jednoduchá transformace, složitější test, slušná aproximace
- „**K-dop tree**” (Klosowski, Held, Mitchell, 1998)
  - složitější transformace a test, výborná aproximace

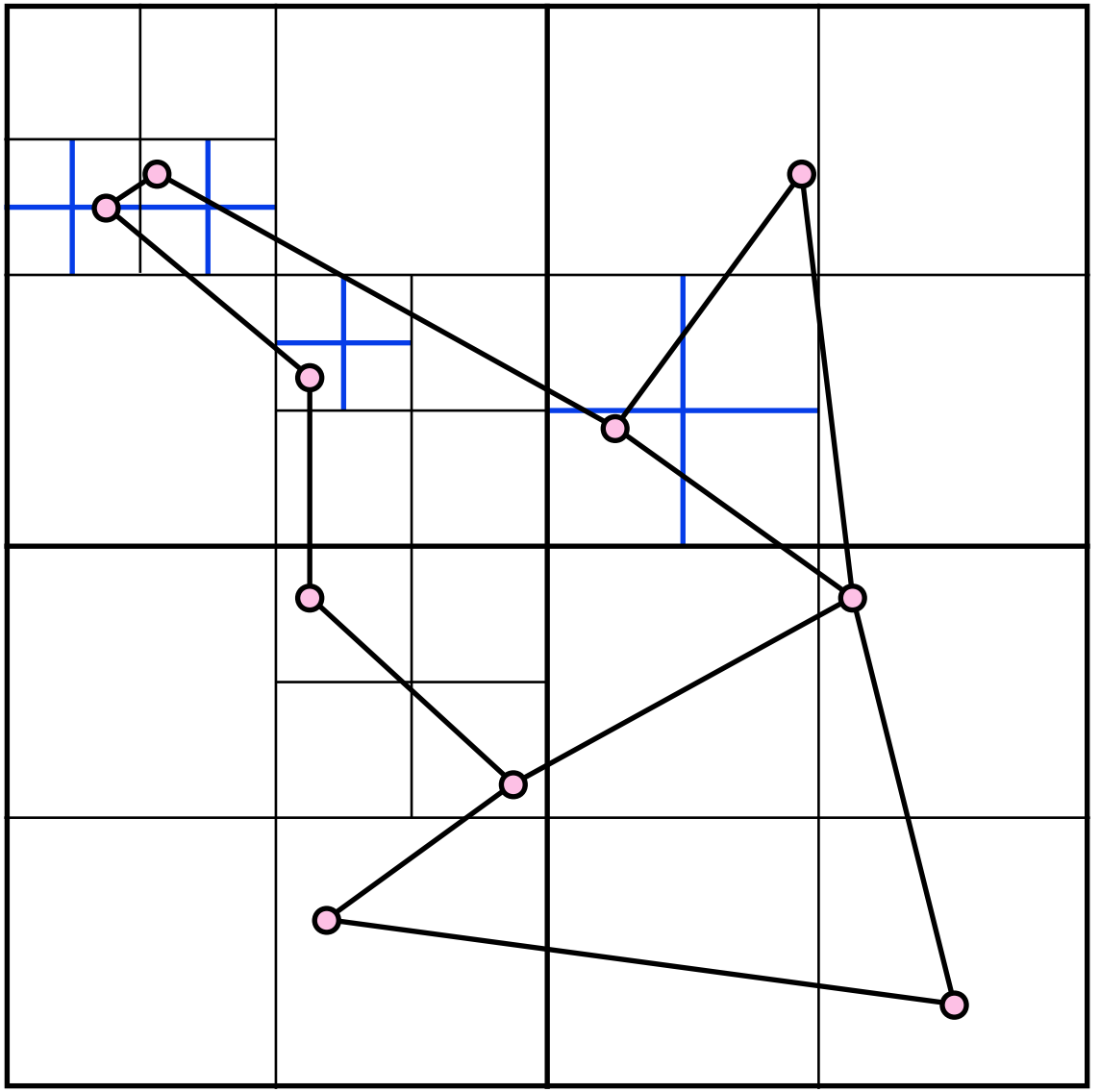
„PM_1 quadtree” (Samet, Webber)

- **reprezentace <u>polygonální mapy</u>**
- **maximálně <u>jeden vrchol/list</u>**
- **list může obsahovat několik hran pouze tehdy, mají-li <u>společný vrchol v daném listu</u>**

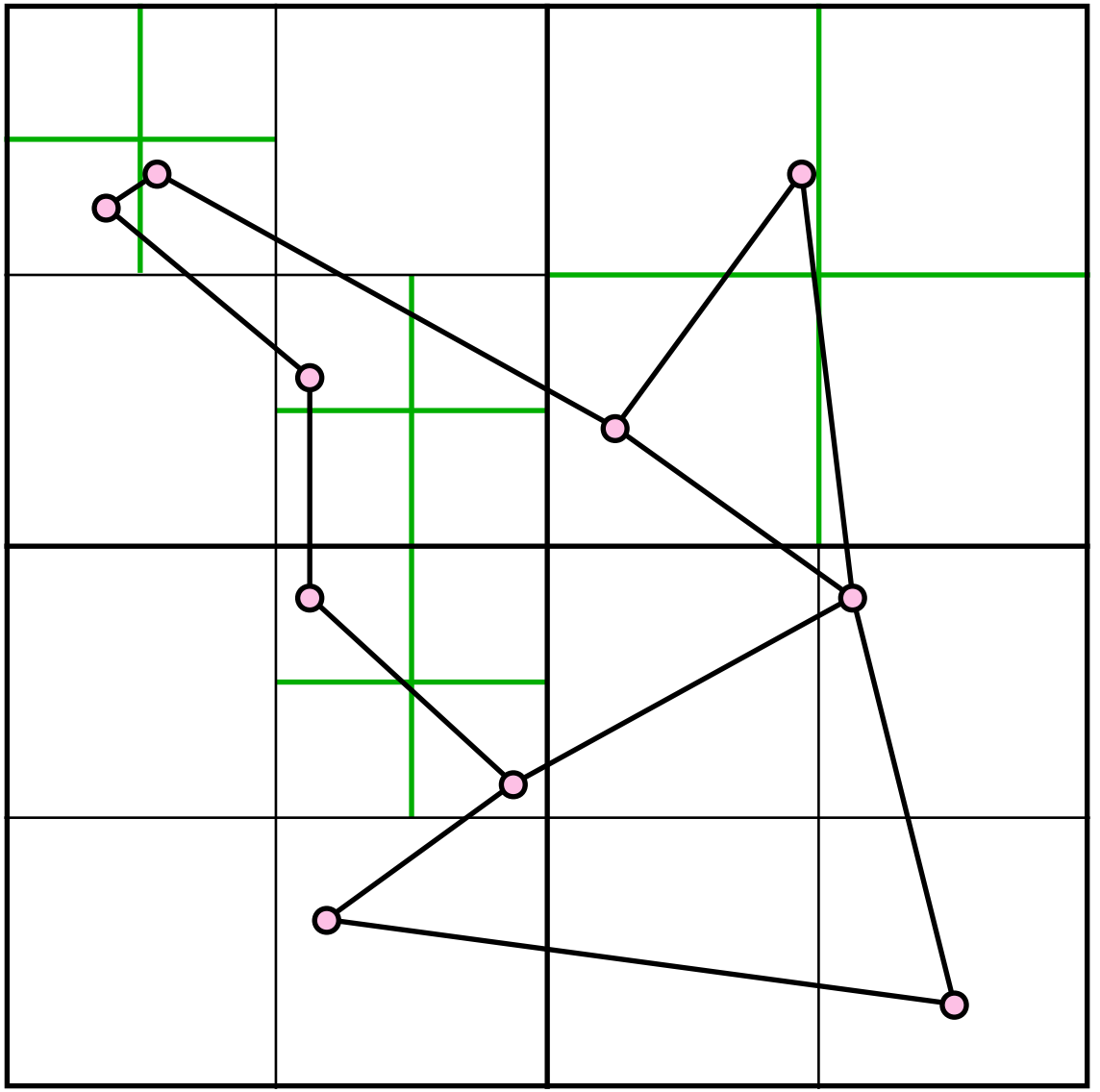


„PM_2 quadtree”

- **reprezentace <u>polygonální mapy</u>**
- **maximálně <u>jeden vrchol/list</u>**
- **list může obsahovat několik hran pouze tehdy, mají-li <u>společný vrchol</u>**



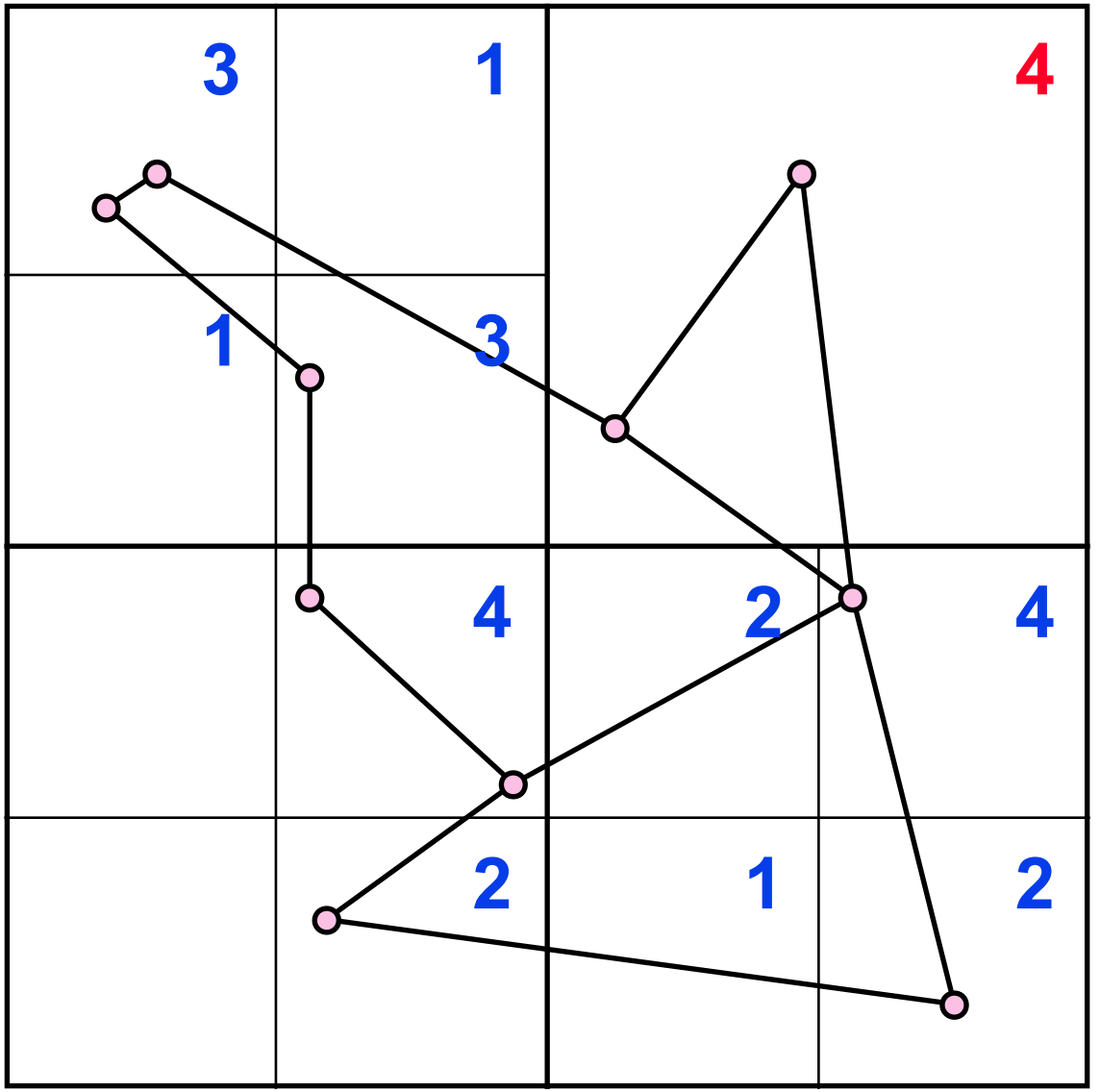
„PM_3 quadtree”



- **reprezentace <u>polygonální mapy</u>**
- **maximálně <u>jeden vrchol/list</u>**
- **list může obsahovat <u>libovolný počet hran</u> (“PR quadtree” s hranami)**

„PMR quadtree”

- **reprezentace <u>polygonální mapy</u>**
- **maximálně N <u>hran/list</u>**
- **list může obsahovat <u>odkazy na několik vrcholů</u> (max. 2N)**



N = 4

Směrový průchod hierarchií

- průchod daty v pořadí daném určitým **směrovým vektorem**
  - výpočet viditelnosti (zepředu-dozadu nebo opačně) v rovnoběžném promítání
  - „plane sweep" průchod („zametání")
- průchod daty **od daného středu**
  - výpočet viditelnosti (konfiguračních faktorů, ..) v perspektivní projekci
  - hledání **N** nejbližších objektů
- výpočet **průsečíku paprsku s 3D scénou**, atd.

Předpoklady

- dekompozice prostoru má **hierarchický charakter**
  - potomci nemusí nutně tvořit zjemnění svého předka (např. „strip tree")
- efektivní výpočet **minimální vzdálenosti (kritéria) každé buňky ... d(B)**
  - vzdálenost od referenční roviny (sweep) nebo od středu zpracování (perspektiva)
  - kritérium nemusí být infimem vzdálenosti (ale musí to být dolní mez)

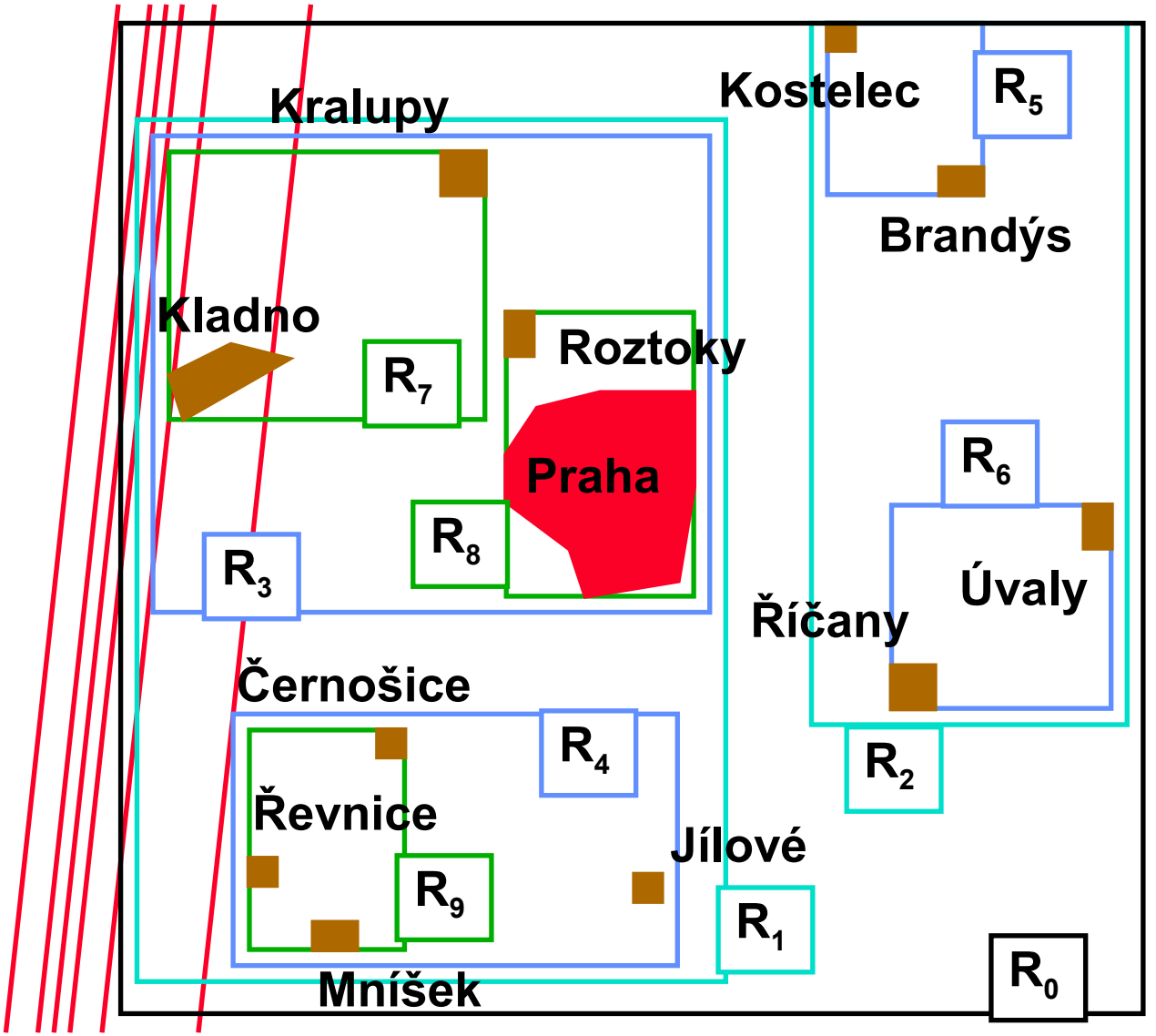
Algoritmus

- pomocná datová struktura **H** (např. halda)
  – rychlé operace **Min**, **DeleteMin**, **Insert**

1. na haldu **H** se uloží **kořen stromu**
   – klíčem pro třídění je kritérium **d(B)**
2. je-li **Min(H) objektem**, zpracuje se a odstraní se z haldy
3. je-li **Min(H) buňkou hierarchie**, nahradí se všemi svými potomky
4. kroky 2 a 3 se opakují až do vyprázdnění **H** nebo zpracování požadovaného počtu objektů

Příklad I

- $\{R_0\} \rightarrow \{R_1, R_2\}$
- $\{R_1, R_2\} \rightarrow \{R_3, R_4, R_2\}$
- $\{R_3, R_4, R_2\} \rightarrow \{R_7, R_4, R_8, R_2\}$
- $\{R_7, R_4, R_8, R_2\} \rightarrow \{$Kladno$, R_4,$ Kralupy$, R_8, R_2\}$
- 1. Kladno
- $\{R_4$, Kralupy$, R_8, R_2\} \rightarrow \{R_9$, Kralupy$, R_8,$ Jílové$, R_2\}$



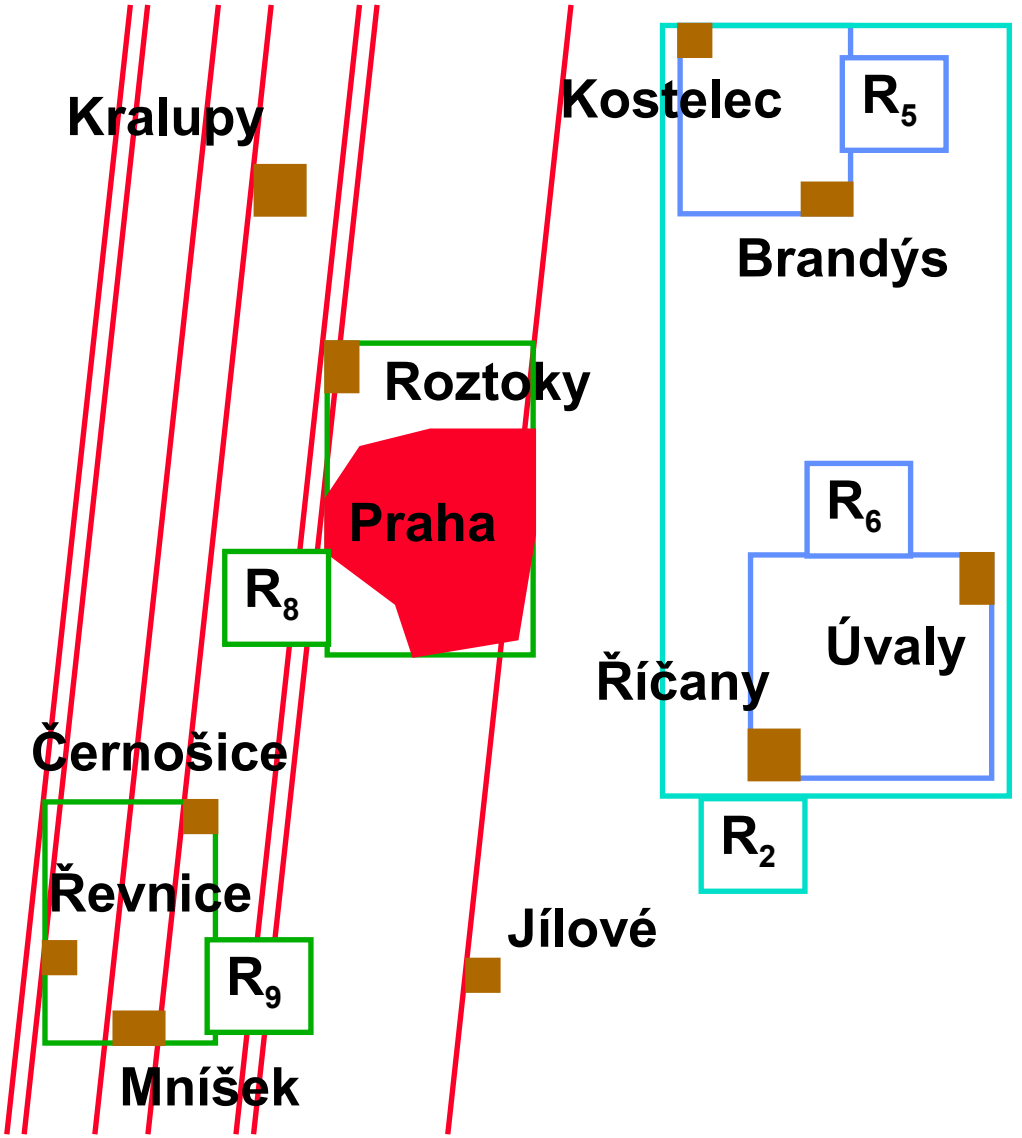
Příklad II

- {R_9,Kralupy,R_8,Jílové,R_2}
  →{Řevnice,Mníšek,
  Kralupy,Černošice,
  R_8,Jílové,R_2}

❷-❺ Řevnice,Mníšek,
Kralupy,Černošice

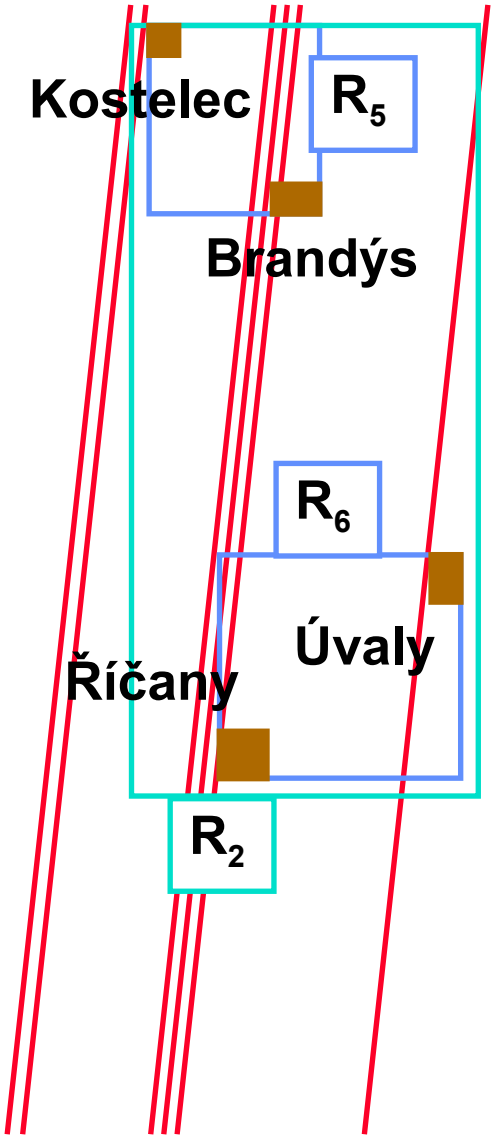
- {R_8,Jílové,R_2}
  →{Roztoky,Praha,
  Jílové,R_2}

❻-❽ Roztoky,Praha,
Jílové



Příklad III

- {R_2}→{R_5,R_6}
- {R_5,R_6}→{Kostelec,R_6,Brandýs}
- ❾ Kostelec
- {R_6,Brandýs}
  →{Brandýs,Říčany,Úvaly}
- ❿-❶❷ Brandýs,Říčany,Úvaly



Konec

Další informace:

- **H. Samet**: ***The Design and Analysis of Spatial Data Structures***, Addison-Wesley, 1990
- **H. Samet**: ***Foundations of Multidimensional and Metric Data Structures***, Morgan Kaufmann, 2006
- **F. Preparata, M. Shamos**: ***Computational Geometry, An Introduction***, Springer-Verlag, 1985