Další varianty R-stromů

3. listopadu 2020

Hilbertův R-strom

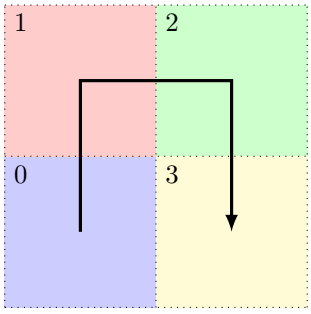
- Je hybridní struktura založená na R-stromu a B^+-stromu.
- Ve skutečnosti je to B^+-strom geometrickými objekty charakterizovanými podle Hilbertovy hodnoty jejich těžiště.
- Struktura je založena na Hilbertově křivce vyplňování prostoru – bylo ukázáno, že Hilbertova dobře zachovává blízkost prostorových objektů.

Hilbertova křivka

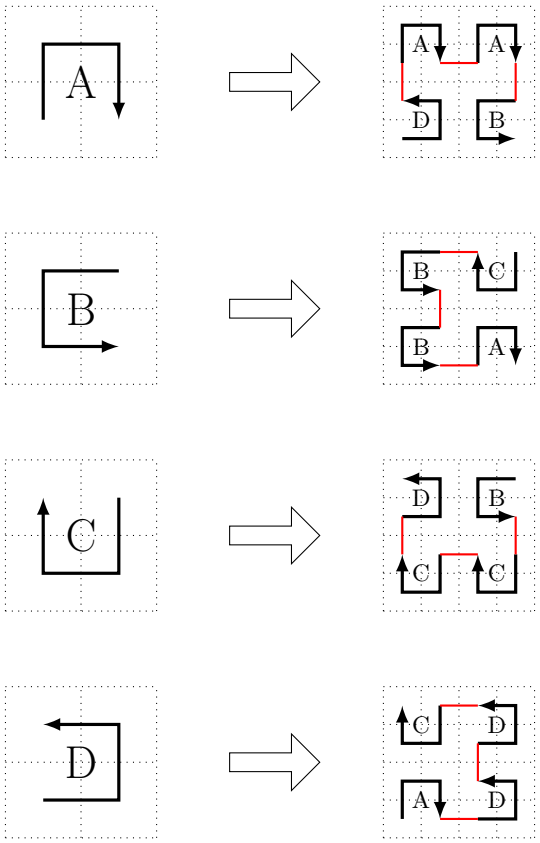
Vysvětlení Hilbertovy křivky. Podívejme se na obr. 1 – vidíme Hilbertovu křivku prvního řádu ($n = 1$), tj. jednoduchou křivku procházejicí prostor rozdělený do čtyř ($2^n \times 2^n$) oblastí, v pořadí 0 (modrý)–1 (červený)–2(zelený)–3(žlutý).

Hilbertovy křivku vyššího řádu $(n+1)$ vytvoříme rekurzivně z křivky nižšího řádu (n) pomocí pravidel na obr. 2. Například, aplikací prvního pravidla v obr. 2 získáme z Hilbertovy křivky prvního řádu Hilbertovu křivku druhého řádu na obr. 3. Všimněme si, že platí, že Hilbertova křivka řádu $n = 2, 3, \ldots$

- se skládá ze čtyř Hilbertových křivek řádu $n - 1$ (patřičně převrácenými);



Obrázek 1: Hilbertova křivka prvního řádu



Obrázek 2: Pravidla pro generování Hilbertových křivek vyšších řádů

- proto taky mapuje prostor rozdělený do čtyřikrát více oblastí;
- že prochází původní oblasti (oblasti křivky řádu $n-1$;) ve stejném pořadí, jako Hilbertova křivka řádu $n-1$.

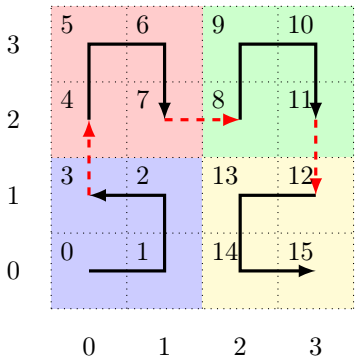
Na obr. 4 vidíme křivky řádů 3, 4 a 5.

Zpět k Hilbertovým R-stromům – vkládání

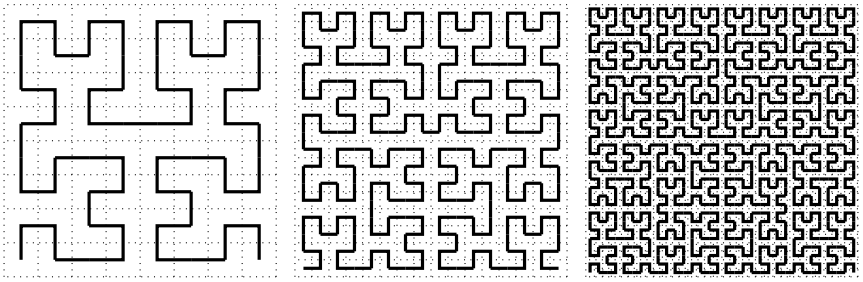
U Hilbertových R-stromů uvažujeme, že známe hranice oblasti, ve které se budou vkládaná data (případně jejich těžiště) vyskytovat. Tuto oblast proložíme dostatečně jemnou $2^n \times 2^n$ mřížkou a oblasti této mřížky namapujeme na Hilbertovu křivku řádu n.

Při vkládání dat do Hilbertova R-stromu:

- Vypočteme H-hodnotu – pozici oblasti, do které padne těžiště, na Hilbertově křivce.



Obrázek 3: Hilbertova křivka druhého řádu



Obrázek 4: Hilbertovy křivky řádu 3, 4 a 5 (zleva doprava)

- Poté vložíme data stejně, jako bychom vkládali do B^+-stromu (uzly tedy oproti R-stromu musí navíc obsahovat H-hodnoty v nelistových uzlech).
- Upravíme obdélníky steně, jako u R-stromu.

Budeme potřebovat algoritmus, který pro argumenty

- souřadnice (x, y) oblasti,
- řád $n \in \mathbf{N}$ Hilbertovy křivky,

vrací H-hodnotu. Tedy například, pro $(x, y) = (2, 1); n = 2$ vrací 13 (viz obr. 3).

Tento algoritmus bude mít rekurzivní povahu, a bude asi takto:

- Pokud $n = 1$, zjistíme H-hodnotu přímo ze souřadnic (x, y):

$$(0,0) \mapsto 0 \qquad (0,1) \mapsto 1 \qquad (1,1) \mapsto 2 \qquad (1,0) \mapsto 3 \qquad (1)$$

- Jinak z (x, y) zjistíme kvadrant, a rekurzivně řešíme pro $n - 1$ a upravené (x, y).

Např. pro $(x, y) = (2, 1); n = 2$, vidíme (viz obr. 3), že se nacházíme ve čtvrtém žlutém kvadrantu, to znamená, že výsledku rekurzivního volání přičteme $3 \cdot (2^{n-1} \cdot 2^{n-1}) = 12$ za tři kvadranty, které mu předchází. To rekurzivní volání, provádíme pro Hilbertovu křivku, která je v tom žlutém kvadrantu. Musíme upravit (x, y):[1]

$$\begin{aligned} x &\leftarrow 2^{n-1} - 1 - y & y &\leftarrow 2^n - 1 - x \\ &= 2 - 1 - 1 = 0 & &= 3 - 1 - 2 = 1 \end{aligned}$$

Dle (1) tedy dostáváme 1. V součtu s tou 12 dostáváme očekávaných 14.

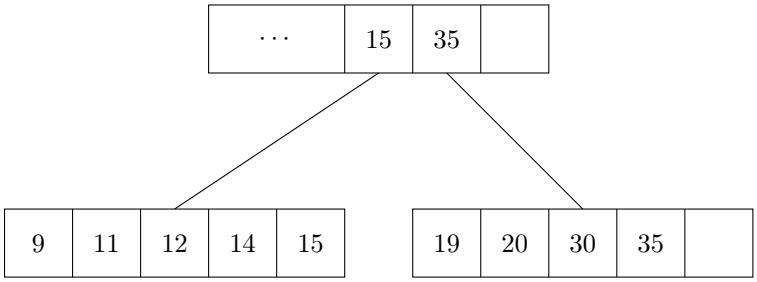
Poznámka: Případně můžeme proceduru ještě víc zjednodušit, když budeme uvažovat i $n = 0$, kdy křivka pokrývá jen jednu oblast, a vrací vždy 0.

Tento algoritmus domyslete a zpracujte jako domácí úkol.

[1] Toto bude vypadat pro každý kvadrant jinak.

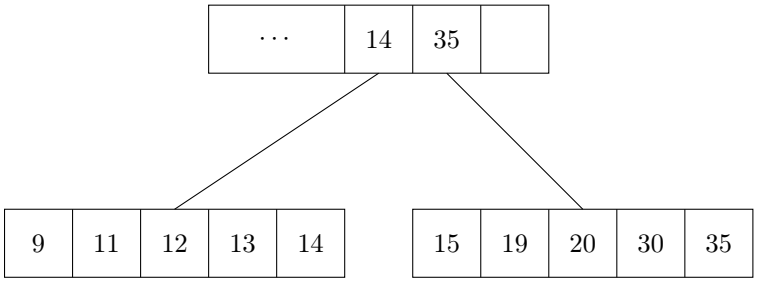
Přeplnění uzlu

Díky organizaci přes H-hodnoty může Hilbertův R-strom využívat stejné finty jako B^*-stromy. Tj. nelámat hned, jakmile dojde k přeplnění uzlu, ale vložíme do sousedního uzlu. Pokud je i sousední uzel plný, lámeme přeplněný uzel i jeho plného souseda na tři uzly. Viz grafický příklad:



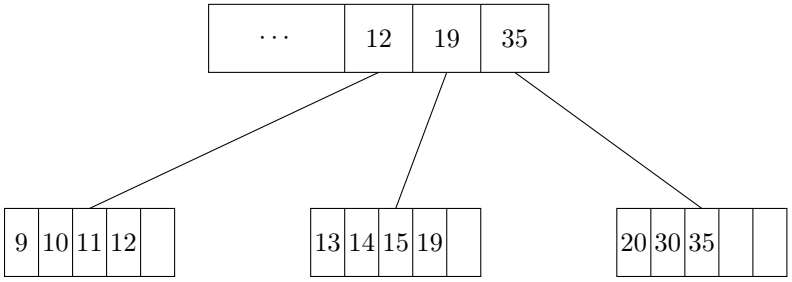
přidáváme data s H-hodnotou 13

Výsledek:



přidáváme data s H-hodnotou 10

Výsledek:



Vyhledávání v Hilbertově stromu

Vyhledávání probíhá stejným způsobem jako v R-stromech. H-hodnoty jsou tedy ignorovány a používají se pouze MBR stejným způsobem jako v R-stromech.

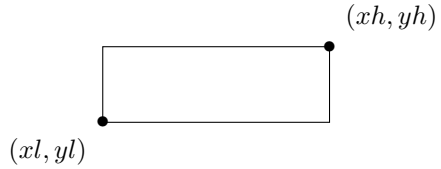
Lineární split

Ukážeme si ještě jeden lineární algoritmus pro distribuci objektů přeplněného uzlu do dvou množin.

- Primárním kritériem tohoto algoritmu je rozdělit objekty mezi dva uzly co nejrovnoměrněji.
- Druhým kritériem je minimalizace jejich překrytí.
- Třetím kritériem je minimalizace celkového pokrytí.

Aby se minimalizovalo překrývání, snažíme se dát všechny obdélníky v přeplněném uzlu co nejdál od sebe, k okrajům MBR přeplněného uzlu.

Nech je každý obdélník označený jako (xl, yl, xh, yh):



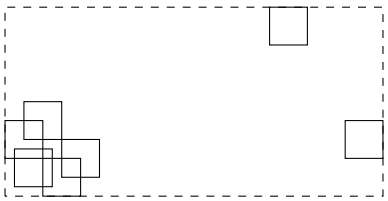
MBR přeplněného uzlu N je označen $R_N = (L, B, R, T)$.

Tento algoritmus používá čtyři listy $\text{LIST}_L, \text{LIST}_B, \text{LIST}_R, \text{LIST}_T$, které obsahují obdélníky z N, které jsou blíže odpovídající hranici, než opačné (opačné hranice jsou uvažovány přirozeně: L–R, B–T).

Popis v algoritmu je v pseudokódu 1:

Můžeme si snadno všimnout, že každý obdélník v N bude přítomen buďto v seznamu LIST_L nebo LIST_R. Totéž platí pro LIST_B a LIST_T. Rozhodnutí, za udělat split podle horizontální nebo vertikální osy závisí na rozložení obdélníků (tomu odpovídají řádky v pseudokódu, které porovnávají maxima velikostí seznamů). V případě remízy, se algoritmus rozhodne podle kritéria překryvu.

Tento algoritmus může mít problém v případě, že většina obdélníků v N tvoří shluk a několik dalších jich je odlehlých. V pak velikosti seznamů budou velmi nerovnoměrné. Viz obr:



Tento problém je obvykle možno řešit odebráním a a znovu-vložením uzlů, které odpovídají těm odlehlým obdélníkům, podobně jako v R*-stromu.

$\text{LIST}_L \leftarrow \text{LIST}_B \leftarrow \text{LIST}_R \leftarrow \text{LIST}_T \leftarrow \emptyset$;
for *rectangle* $S = (xl, yl, xh, yh)$ *in* N *with* $R_N = (L, B, R, T)$ **do**
  if $xl - L < R - xh$ **then**
    $\text{LIST}_L \leftarrow \text{LIST}_L \cup S$
  else
    $\text{LIST}_R \leftarrow \text{LIST}_R \cup S$
  end
  if $yl - B < T - yh$ **then**
    $\text{LIST}_B \leftarrow \text{LIST}_B \cup S$
  else
    $\text{LIST}_T \leftarrow \text{LIST}_T \cup S$
  end
end
if $\max(|\text{LIST}_L|, |\text{LIST}_R|) < \max(|\text{LIST}_B|, |\text{LIST}_T|)$ **then**
  Split the node along the x direction.
else
  $\max(|\text{LIST}_L|, |\text{LIST}_R|) > \max(|\text{LIST}_B|, |\text{LIST}_T|)$
Split the node along the y direction. **else**
  if $\text{overlap}(\text{LIST}_L, \text{LIST}_R) < \text{overlap}(\text{LIST}_B, \text{LIST}_T)$ **then**
    Split the node along the x direction.
  else
    $\text{overlap}(\text{LIST}_L, \text{LIST}_R) > \text{overlap}(\text{LIST}_B, \text{LIST}_T)$
  Split the node along the y direction. **else**
    Split the node along the direction with smallest total coverage.
  end
end

Algorithm 1: Nový lineární split