Skyliny: DS (datová struktura)

16. března 2013

1 Původní algoritmus

Parametry:

- prostor a konstrukční čas $\mathcal{O}\left(n \log^{d-2} n\right)$,
- dotaz v čase $\mathcal{O}\left(\log^{d-1} n\right)$.

2 vEB stromy

Neperzistentní vEB stromy jsou klasický výsledek, jinak můžete kouknout do mé diplomky. Randomizovaná verze se asi lépe nahlédne v *y-fast* triích.

- **Předpokládáme:** word-RAM se slovy délky w, naše univerzum má velikost N kde prvek se vejde do konstantního počtu slov (tedy $w \in \Omega(\log N)$).
- **Všimněte si**: v našem případě N je pouze počet bodů na vstupu, což jsme značili n. Předpokládám že v preprocesingu si složky vstupních vektorů porovnáváním setřídíme a očíslujeme.
- **Čas:** na každou operaci je $\log\log N$ kroků. Podporované operace jsou vkládání a mazání, nejbližší prvek menší/větší než dané číslo. Také lze dělat průchod v $\mathcal{O}(1)$ na krok.
- **Prostor**: deterministicky je $\mathcal{O}(N \log\log N)$ a myslím že se neumí výrazně lépe. Randomizovaně se umí lineární ve velikosti *reprezentované* množiny (pomocí univerzálního hashování).
- **Konstrukce**: ze setříděné množiny jde (skoro určitě) v čase $\mathcal{O}(n)$. Navíc pro naši úlohu umíme v $\mathcal{O}(n)$ i třídit.
- **Závěr**: pro náš případ tedy není jen $\mathcal{O}\left(\sqrt{\log n}\right)$ na prvek, ale tedy dokonce $\mathcal{O}(\log\log n)$. Husté univerzum v problému předchůdce hrozně pomáhá.

3 Perzistentní vEB stromy

V perzistenci jde o to jak efektivně reprezentovat celou historii vývoje datové struktury. Nám bude stačit varianta s *lineární* historií. Následující článek by to měl dělat, bez zhoršení asymptotických vlastností (prostor je tedy lineární v celkovém počtu jednotkových modifikujících operací). Článek jsem myslím četl a byla to pouze relativně přímočará úprava vEB stromů pomocí perzistentních technik.

Timothy M. Chan: Persistent Predecessor Search and Orthogonal Point Location on the Word RAM. odkaz

4 SemiDynamizace

- **Problém**: odpovídat dotaz zda zadaný bod je dominován některým z reprezentované množiny, budu značit predikátem $\mathsf{dom}(S, x)$ (tedy platí když "množina dominuje bod").

- **Předpoklady**: máme statickou strukturu s konstrukčním časem $P(d, n)$, časem dotazu $Q(d, n)$ a prostorem třeba $M(d, n)$.

- **Rozložitelnost:** pokud množinu bodů S rozložíme tak že $\bigcup_i S_i = S$ (ne nutně disjunktně), pak $\mathsf{dom}(S, x) \leftrightarrow \exists i\ \mathsf{dom}(S_i, x)$.

- **Cíl**: získat semidynamickou strukturu, tedy stačí přidat vkládání. Na to je klasická metoda, nejspíš Saxe a Bentley okolo 1978 (myslím si že Overmars a Leeuwen až okolo 1980 přidávali silné a slabé mazání a deamortizaci, odkaz).

- **Výsledek**: pokud se nepletu, pak prostor $\mathcal{O}(M(d, n))$, dotaz $\mathcal{O}(Q(d, n) \cdot \log n)$ a vložení $\mathcal{O}(P(d, n) \cdot \log n/n)$.

5 Dohromady

5.1 Zatím velmi nahrubo (a možná nepravdivě)

Dle mého odhadu bychom měli dostat strukturu s následujícími vlastnostmi:

- lineární prostor (ve velikosti dat, tedy $\mathcal{O}(nd)$ slov),

- konstrukce v čase $\mathcal{O}(nd \log \log n)$,

- dotaz v čase $\mathcal{O}(d \log \log n)$ staticky nebo $\mathcal{O}(d \log n \log \log n)$ dynamicky.

Navíc si myslím že vyextrahovat z té datové struktury skylinu také půjde, nebo minimálně by mělo jít se po konstrukci zeptat na každý vstupní bod...

:-)

5.2 Induktivní konstrukce podrobněji

Zdá se že to zdaleka nepůjde tak jednoduše (mně to od začátku přišlo podezřelé).

- **Předpoklad**: potřebujeme dynamickou a perzistentní (lineární historie) DS(d) pro dotazy. Perzistence je potřeba kvůli prostoru, na perzistenci je potřeba dynamičnost DS($d-1$).
- **Základ**: DS(2) je samotný perzistentní vEB strom. Dynamičnost a perzistence je potřeba pouze kvůli indukci. Prostor je $\mathcal{O}(n)$, čas $\mathcal{O}(\log\log n)$ na operaci.
- obecný indukční krok bude potřeba udělat bez asympotické notace, s pojmenovanými konstantami
- DS(3) je sekvence upravovaných DS(2), indexovaných dalším vEB stromem dle přidané souřadnice.
  - Statickou DS(3) tedy máme v prostoru $\mathcal{O}(n)$ a čase $\mathcal{O}(\log\log n)$ na dotaz.
  - Po dynamizaci se dostaneme na časy pro dotaz $\mathcal{O}(\log n \log\log n)$ a pro vložení $\mathcal{O}(\log n)$. Prostor by měl být $\mathcal{O}(n)$ plus $\mathcal{O}(\log n)$ za každou perzistentní modifikaci. (Bohužel. Možná že to půjde vylepšit nějakou jinou perzistentně-dynamizační konstrukcí, ty koncepty k sobě mají blízko.)
- DS(4) je sekvence upravovaných DS(3), indexovaných dalším vEB stromem dle přidané souřadnice.
  - Statická DS(4) má dotaz v čase $\mathcal{O}(\log n \log\log n)$, čas konstrukce a prostor je $\mathcal{O}(n \log n)$. Srovnání: původně byl prostor $\mathcal{O}\left(n \log^2 n\right)$ a čas dotazu $\mathcal{O}\left(\log^3 n\right)$, tedy už máme slušné zlepšení (pokud někde není chyba).
  - Po dynamizaci se dostaneme na časy pro dotaz $\mathcal{O}\left(\log^2 n \log\log n\right)$ a pro vložení $\mathcal{O}\left(\log^2 n\right)$. Prostor by měl být $\mathcal{O}(n \log n)$ plus odhaduju $\mathcal{O}\left(\log^2 n\right)$ za každou perzistentní modifikaci.
- Zdá se že náš přímočarý postup neškáluje v d tak dobře jak jsme předpokládali...