11. cvičení

Úloha 1 (Nejhorší případ pro k-d strom)
Mějme 2-d strom, tedy binární strom, kde rozdělujeme rovinu střídavě podle os x a y. Ukažte, že dvourozměrný intervalový dotaz (na počet bodů v obdélníku) může trvat až $\Omega(\sqrt{n})$, konkrétně najděte množinu bodů uloženou ve stromě a dotaz, který bude trvat takto dlouho.
Bonus: zkuste toto rozšířit na obecné k-d stromy s časovou složitostí $\Omega(n^{1-\frac{1}{k}})$.

Úloha 2 (Hledání minima a mazání v k-d stromě)
Máme k-d strom. Navrhněte, jak na něm implementovat operaci nalezení nejmenšího prvku v dané dimenzi. (Tedy funkci FindMin(d), která nalezne prvek, jehož d-tá souřadnice je minimální.)
Dále navrhněte implementaci operace Delete. Pro jednoduchost můžete předpokládat, že žádné dva body spolu nesdílejí žádnou souřadnici.
(V obou případech neřešte časovou složitost.)

Úloha 3 (Nejbližší soused a k-NN)
Máme k-d strom. Navrhněte, jak na něm implementovat operaci nalezení nejbližšího souseda, a operaci nalezení m nejbližších sousedů.
Hint: pro m-NN se může hodit mít k dispozici haldu.

Úloha 4 (Zrychlení intervalových stromů)
Mějme zjednodušený dvourozměrný intervalový strom, tedy binární strom podle souřadnice x, který má v každém vrcholu pole bodů z daného x-ového intervalu seřazené podle y. Dvourozměrné intervalové dotazy (na počet bodů v obdélníku) zde trvají $\mathcal{O}(\log^2 n)$, protože potřebujeme binárně vyhledávat v $\mathcal{O}(\log n)$ polích. Ukažte, že přidáním (lineárně mnoha) pointerů z každého pole na správná místa v obou polích v synech můžeme složitost snížit na $\mathcal{O}(\log n)$.
(Pro d-rozměrný strom pak dostaneme $\mathcal{O}(\log^{d-1} n)$.)

Bonusy

Úloha 5 (Něco „praktičtějšího“)
Při hashování často počítáme modulo, ale to se přeloží do strojového kódu jako operace `idiv`, která je hodně drahá (latence pro 64 bitová čísla může být, dle procesoru, klidně i řádově desítky cyklů). Existují ale speciální prvočísla, tzv. Mersennova, pro která to jde rychleji. Ta mají tvar $p = 2^s - 1$.
Zkuste naimplementovat modulo Mersennovo prvočíslo jen pomocí bitových operací (bitshift, bitwise and/or), sčítání, odčítání a porovnávání (obecně rychlých operací).