13. cvičení

Datové struktury I, 3. 1. 2023 `https://iuuk.mff.cuni.cz/~chmel/2223/ds1/`

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 $\Omega(n^{1-\frac{1}{k}})$.

Řešení
Vezmeme množinu $\{(i,i) : 1 \leq i \leq n\}$ a budeme se ptát na dotaz $\{0\} \times \mathbb{R}$. Pro jednoduchost mějme $n = 2^t - 1$. V každém x-vrcholu jdeme doleva, ale v každém y-vrcholu musíme jít oběma směry. Strom má hloubku $t \approx \log n$, a v každém druhém kroku zdvojnásobíme počet podstromů, na které se musíme podívat. Celková složitost tedy je $\Omega(2^{(\log n)/2}) = \Omega(\sqrt{n})$.
Rozšíření je analogické, jenom budeme mít k-tice $(i, \ldots, i)$ a dotaz bude $\{0\} \times \mathbb{R}^{k-1}$.

Úloha 2 (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.

Řešení
Máme-li bod x, začneme jako bychom hledali x. Jakmile dorazíme do listu, zapamatujeme si poslední vrchol jako nejbližší. Budeme se v rekurzi vracet, a v každém vrcholu se podíváme, jestli vrchol není bližší, a jestli není šance, že v druhém podstromě může být bližší vrchol. (Vzdálenost nejbližšího bodu druhého obdélníka musí být menší než vzdálenost k nejbližšímu vrcholu.) Pokud šance je, podíváme se i do druhého podstromu.
Zlepšení: to samé, jenom si nejbližší vrcholy ukládám do haldy a beru vrcholy shora, abych měl aspoň nějaké omezení.

Úloha 3 (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 $\mathcal{O}(\log^{d-1} n)$.)

Řešení
Pro každý vrchol v a jeho syna w povede z každého prvku pole ve vrcholu pointer na stejný prvek nebo největší menší v poli w. Binární vyhledávání poté provedeme jen v největším poli v kořeni. Pro nalezení odpovídajících hranic v poli v synovi pak stačí jen použít pointery a příp. se posunout o jeden prvek vedle. Během vyhledávaní ve stromě řazeném podle x-ové souřadnice tak jen s konstantním zpomalením udržujeme hranice v poli v aktuálním vrcholu. Této metodě se říká Fractional cascading a obecně umožňuje i propojování polí se stejným řazením ale různými prvky.

Úloha 4 (Předbíháme přednášku)
Na adrese `https://deadlockempire.github.io/` najdete hru Deadlock Empire. Cílem hry je krokovat vícevláknový kód, aby došlo k nějaké nechtěné situaci jako souběžné vykonání kritické sekce, deadlock a podobně.