**ŽILINSKÁ UNIVERZITA V ŽILINE**  
 Fakulta riadenia a informatiky



**ALGORITMY A ÚDAJOVÉ ŠTRUKTÚRY 2**Semestrálna práca

**Vypracovala**: Bc. Dávid Zimen **Vyučujúci:** Ing. Peter Jankovič, PhD.  
**Študijný program:** aplikovaná informatika **Akademický rok:** 2023/2024 **Semester:** zimný  
**Ročník**: prvý, inžiniersky stupeň

**Obsah**

[Úvod 3](#_Toc150124957)

[1 Nákres systému 3](#_Toc150124958)

[2 Quad strom 4](#_Toc150124959)

[2.1 Funkcia intervalového vyhľadávania 4](#_Toc150124960)

[2.2 Zlepšenie funkcií 4](#_Toc150124961)

[2.3 Zdravie stromu 4](#_Toc150124962)

[2.3.1 Funkcia na výpočet zdravia 5](#_Toc150124963)

[2.3.2 Optimalizačná funkcia 5](#_Toc150124964)

[2.3.3 Testovanie optimalizačnej funkcie 6](#_Toc150124965)

[2.4 Zložitosti operácií 7](#_Toc150124966)

[3 Práca so súbormi 7](#_Toc150124967)

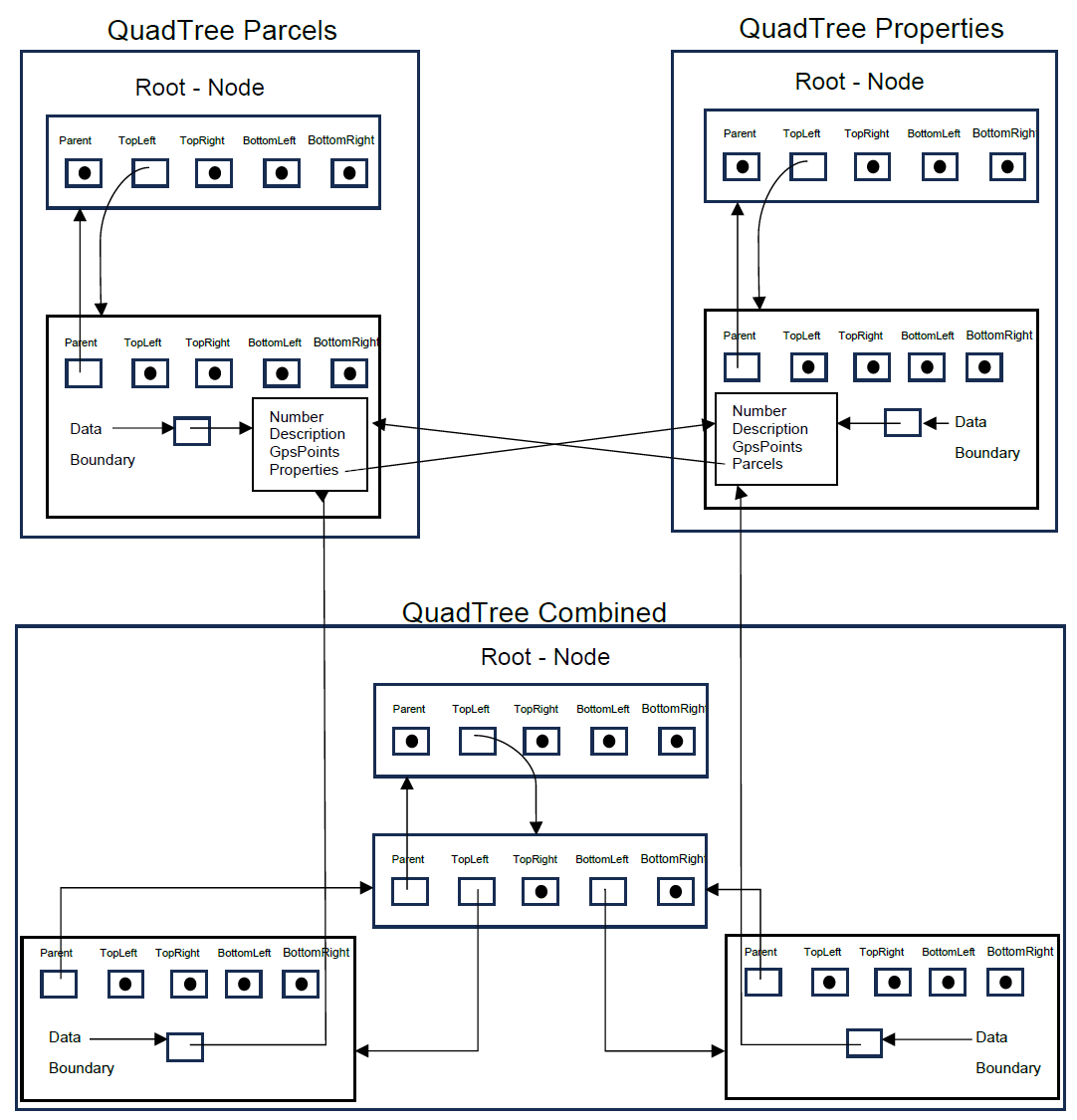
# Úvod

V nasledujúcej dokumentácii si zhrnieme semestrálnu prácu 1, ktorej cieľom bolo vytvoriť systém pre geodetov na ukladanie dát o parcelách a nehnuteľnostiach.

Daný systém je potrebné implementovať pomocou údajovej štruktúry Quad strom. Štruktúru je nutné samostatne naimplementovať a predstaviť do nej aj určité vylepšenia operácia vymaž a vlož, funkciu na výpočet zdravia štruktúry a následne optimalizáciu na základe hodnoty zdravia.

Taktiež je potrebné zabezpečiť ukladanie a opätovné načítanie dát zo súboru vo formáte CSV.

# Nákres systému



Obrázok 1.**Chyba! Na použitie štýlu 0 na text, ktorý sa má zobraziť na tomto mieste, použite kartu Domov.**. Nákres systému

Systém ako celok bude pozostávať z troch Quad stromov. V jednom budú uložené dáta ohľadom parciel. V dalšom budú uložené dáta týkajúce sa nehnuteľností a na koniec bude jeden spoločný strom, kde bude možné ukladať dáta nezávislá na ich povahe.

Dáta sa do jednotlivých kvadrantov stromu delia na základe ich dvoch ohraničujúcich GPS pozícií. Prvá pozícia reprezentuje ľavú hornú súradnicu a druhá predstavuje pravú dolnú súradnicu.

Automatické naplnenie parciel ku nehnuteľnostiam bude prebiehať tak, že po uložení do stromu nehnuteľností sa zavolá funkcia hľadaj podľa ohraničujúcich bodov na strome s parcela a podľa intervalového vyhľadávania nájde všetky parcely, ktoré sa pretínajú s pridanou nehnuteľnosťou. Vrátené parcelu sa následne uložia do zoznamu ku nehnuteľnosti. To isté platí opačne ku vloženej parcele.

Do spoločného stromu sa pridávajú položky v momente keď sa vložia do ich príslušných stromov.

# Quad strom

V nasledujúcich podkapitolách si popíšeme spôsob implementácie vylepšení a testovania Quad stromu.

## Funkcia intervalového vyhľadávania

Začína od koreňa stromu a funguje nasledovne.

1. Zober vrchol
2. Pozri, či sa vyhľadávaná súradnica pretína s potomkami
   1. Ak sa pretína, tak pridaj potomka do zásobníka prehľadávania
3. Prezri dáta vo vrchole
   1. Ak sa pretína s dátami, tak pridaj to zoznamu výsledkov
4. Ak nie je zásobník prázdny, choď na krok 1
5. Vráť zoznam výsledkov

## Zlepšenie funkcií

Na samostatné zlepšenie funkcií bolo použité rozdelenie zoznamu duplicít vo vrchole na dva zoznamy. Jeden, do ktorého sa budú ukladať dáta, ktoré sa môžu potencionálne pri delení vrcholu dostať na nižšiu úroveň. V druhom budú dáta, ktoré sa už nemôžu dostať nižšie.

Pri vkladaní to znamená, že keď dôjde ku situácii, že vrchol sa bude deliť na menšie, tak pri delení nebude nutné prechádzať cez všetky dáta vrcholu a overovať, ktoré všetky sa môžu posunúť na nižšiu úroveň.

Strom, ktorý disponuje vylepšením budeme volať AdvancedQuadTree a bez vylepšenia bude ClassicQuadTree.

## Zdravie stromu

V Quad strome je možné zbierať rôzne údaje ohľadom jeho kondície. Pre nás najdôležitejšie, ktoré sa neskôr použijú na výpočet zdravia sú:

#### DataBalanceFactorX

Celé číslo, ktoré vyjadruje, na ktorý smer sú dáta viac naklonené po x-ovej osi. Kladná hodnota vyjadruje, že dáta sa sústreďujú na ľavú stranu, záporná naopak znamená sústredenia viac vpravo.

#### DataBalanceFactorX

Celé číslo, ktoré vyjadruje, na ktorý smer sú dáta viac naklonené po y-ovej osi. Kladná hodnota vyjadruje, že dáta sú rozložené smerom hore, záporná naopak znamená rozloženie smerom nado.

#### DivisibleDataSize

Udáva počet vložených dát, ktoré by sa potencionálne mohli uložiť na nižšiu úroveň ale nastavená maximálna výška stromu im to nedovolí.

#### Depth

Aktuálna hĺbka najhlbšie položeného vrcholu v strome.

Tieto metriky sa získavajú asynchrónne v každom potomkovi koreňa, keď prídu výsledky z jednotlivých podstromov, tak sa vyhodnotia a spoja sa do jednej celkovej metriky.

### Funkcia na výpočet zdravia

Táto funkcia vráti hodnotu v intervali <0; 1>, pričom 0 je najmenej zdravý strom a 1 predstavuje najzdravší strom. Jednotlivé váhy ako idú v poradí v rovnici sú 0,4; 0,4; 0,1; 0,1. To znamená že najväčší dôraz sa dáva na rovnomerné rozloženie dát v strome. Menší dôraz sa potom dáva na to koľko dát sa dá posunúť nižšie a na výšku stromu.

### Optimalizačná funkcia

Na základe získaných metrík sa nasledovne spustí optimalizačná funkcia, ktorá funguje na jednoduchých princípoch.

Nová maximálna výška sa nastaví na hodnotu PotentialDepth, ktorá vyjadruje ako najhlbšie môže ísť nejaký prvok stromu.

Následne sa pomocou balančných faktorov vyrátajú nové súradnice ohraničujúce strom. Tieto súradnice sa oproti pôvodným môžu vľavo a dole zmenšovať. Vpravo a hore potom zväčšovať a to preto aby používateľ mohol do stromu stále vkladať dáta v takých rozsahoch, ako pôvodne zamýšľal.

Potom si vypočítame percento zväčšenia pre jednotlivé body a to podľa toho ako moc je vzdialený najbližší bod. Čím je najbližší bod ďalej, tým je percento zväčšenia vyššie.

Ak je strom naklonený doprava, tak sa zväčší pravá x-ová súradnica stromu o vypočítané percento. Ak je naklonený doľava, tak sa ľavá x-ová zmenší.

Analogicky to následne platí pre náklony smerom nahor a nadol.

Tieto zmeny veľkostí súradníc ohraničujúcich strom sa nevykonajú, ak sú jednotlivé balančné faktory menšie ako hodnota 0,1. Daná hodnota zabezpečuje, že ide o veľmi malá vychýlenie a zmena súradníc by mohla mať v celkovom merítku opačný efekt.

### Testovanie optimalizačnej funkcie

Optimalizačnú funkciu sme otestovali nasledovným spôsobom. Stromom sme na začiatok nastavili relatívne malú maximálnu hĺbku a generátor sme upravili, tak aby generované dáta spadali viac na jednu stranu, nech sú zlé balančné faktory.

Následne sme stromy naplnili na začiatočnú kapacitu 50 000 prvok. Pre tejto kapacite sme spustili 10 000 operácií vlož, vymaž a vyhľadaj a výsledky zaznamenali do tabuľky aj spolu so zdravým stromu. Tie sme spriemerovali a zapísali hodnoty do tabuľky 1.

Tabuľka Priemerná doba trvania 10 000 operácii pri zlom zdraví

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priemery neoptimalizovane [ms] | | | | |
|  | Health | Find | Insert | Delete |
| ClassicQuadTree | 58,608 | 288 | 102,7 | 33,3 |
| AdvancedQuadtree | 58,608 | 279,1 | 77,1 | 30,5 |

Obrázok 2.1 Graf ku tabuľke 1

Následne sme nad stromami zavolali optimalizačnú funkciu a znova zopakovali 10 000 operácii nájdi, vlož a vymaž. A dostali sme nasledovné výsledky.

Tabuľka Priemerná doba trvania 10 000 operácii pri optimalizovanom zdraví

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Priemery optimalizovane [ms] | | | | |
|  | Health | Find | Insert | Delete |
| ClassicQuadTree | 94,328 | 145,8 | 52,8 | 23,4 |
| AdvancedQuadtree | 94,328 | 142,9 | 36,3 | 22,1 |

Obrázok 2. Graf ku tabuľke 2

S vyššie uvedených grafov vyplýva skutočnosť, že optimalizačná funkcia splnila očakávania a pri množstve 50 000 prvkov sa vyhľadávanie zrýchlilo takmer dvojnásobne. A podľa očakávania sa potvrdil aj vplyv na funkciu vloženia, kde pri optimalizovaných aj neoptimalizovaných stromoch, je vidno celkom rozdiel v rýchlosti, pričom vyhľadávanie a mazanie neprináša takmer žiadny rozdiel medzi AdvancedQuadTree a ClassicQuadTree.

## Zložitosti operácií

Pri Quad strome vo všeobecnosti platí, že nedokážeme presne určiť zložitosti jednotlivých operácii, vďaka tomu, že nikdy nevieme na koľko presne uzlov sa pri vkladaní alebo mazaná dát rozdelí.

Najbližšie ku hrubému odhadu majú logaritmické zložitosti, keďže sa jedná o stromovú štruktúru a na základe hraníc sa vieme pustiť jednou zo štyroch ciest. V tom prípade by mali všetky operácie logaritmickú priemernú zložitosť, čiže log n\*m, kde n je počet dát v uzle a m je počet uzlov v strome. Ale ako už bolo spomenuté vyššie, zložitosť sa nedá vyčísliť úplne presne.

# Práca so súbormi

Pre zápis do súboru vytvorená vlastná funkcia, ktorá berie ako parameter meno súboru, typ triedy, ktorá sa bude konvertovať a dáta. Z typu triedy sa pomocou reflexie získa zoznam atribútov z ktorých sa vytvorí hlavička. Následne sa potom každému objektu zo zoznamu, taktiež pomocou reflexie získa každý potrebný atribút a oddelia sa separátorom a zapíšu do súboru.

Čítanie funguje na rovnakom princípe, a to takom, že z typu triedy sa vytvorí inštancia a následne sa jej podľa hlavičky nastavujú jednotlivé atribúty. Načítanie vráti zoznam inštancií objektov danej triedy.

Konkrétne parcely a nehnuteľnosti sa do súborov zapisujú nasledovne:

1. Zober všetky parcely/nehnuteľnosti
2. Ich gps pozície postupne oddeľ do samostatného zoznamu
3. Zapíš parcely/nehnuteľnosti a ich pozície do oddelených súborov

Načítanie potom prebieha takto:

* + - 1. Načítaj parcely/nehnuteľnosti a prislúchajúce pozície
      2. Spáruj ich v takom poradí v akom boli načítane
      3. Ulož do QuadStromu