A picture containing diagram

Description automatically generated**ŽILINSKÁ UNIVERZITA V ŽILINE**

**FAKULTA RIADENIA A INFORMATIKY**

**Algoritmy a Údajové Štruktúry 2**

Semestrálna práca 1 – Kataster

Andrej Brescher 5ZIS22 2023/2024

**Návrh údajovej štruktúry**

Údajová štruktúra má generický parameter „TKey“, ktorý hovorí o tom, aký typ dát sa bude ukladať do štruktúry. Tento typ musí implementovať interface „IQTData“, jeho metódy CompareTo a CompareIntersect. Metóda CompareTo porovná, či do uzla, ktorý je parametrom tejto metódy sa daný objekt zmestí a metóda CompareIntersect porovná či sa pozícia objektu pretína so vstupnými súradnicami. Ďalej treba implementovať metódu EqualsTo, v ktorej sa porovná sekundárny kľuč so sekundárnym kľúčom vstupného parametra.

Štruktúra podporuje vkladanie, mazanie, vyhľadávanie objektov, zmenu výšky stromu a prípadnú optimalizáciu štruktúry.

**Zdravie stromu a jeho optimalizácia**

Zdravie stromu a následná optimalizácia sa počíta v cykle maximálne tri krát. Najskôr sa zistí v ktorom synovi koreňa je najviac a v ktorom najmenej prvkov. Následne sa spraví pomer prvkov v tomto maxime/minime voči ostatným. Ak pomer max/ostatné je väčší ako jedna, tak podľa toho, ktorý uzol obsahuje toto maximum sa zmení rozmer stromu alebo ak pomer min/ostatné je menší ako jedna, tak sa takisto zmení rozmer stromu. Ak sa uznalo za vhodné zmeniť rozmer stromu podľa výpočtu, tak sa tak uskutoční a to v každej iterácií, v prvej sa zväčši plocha stromu o 1/8 pôvodnej veľkosti, v druhej iterácií o 1/16 a v tretej o 1/32.

Následne sa prejde na výpočet optimálnej výšky stromu, a to tak, že sa vypočíta priemerný počet prvkov v listoch ktoré sú na úrovni maximálnej výšky stromu. Ak je tento pomer väčší ako 1, tak sa spraví logaritmus tohto priemerného počtu a pripočíta sa k pôvodnej výške. A naopak, ak je tento pomer menší ako 0,3, tak sa výška stromu zmenší o „priemerný počet prvkov v listoch \* aktuálna výška“.

Ak sa zmenila rozloha stromu, tak sa vytvorí nový strom s novou rozlohou a vložia sa do neho všetky prvky z aktuálneho stromu a prepíše sa koreň starého stromu na nový. Ak sa iba zmenila výška stromu, tak sa pomocou metódy „ChangeHeight“ nastaví na novú výšku. A ak nenastala žiadna zmena, tak cyklus výpočtu zdravia končí.

**Testovanie „chorého“ stromu a výsledok optimalizácie**

Testovanie efektívnosti optimalizácie prebiehalo tak, že sa najskôr naplnil strom o rozlohe 10 000 000 \* 10 000 000 a maximálnou výškou 5, miliónom prvkov o maximálnej rozlohe 25 \* 25. Tieto prvky sa vkladali len do severo-západného syna koreňa, aby sme mohli hovoriť o „nezdravom“ strome. Následne sa vkladal a mazal novo vytvorený prvok, stále vkladaný do oblasti severo-západného syna koreňa a meral sa čas výkonu operácie Insert a Delete. Následne sa zavolala optimalizácia a tak isto a na rovnakej ploche sa vkladal a mazal novovytvorený prvok. Následne sa zmeraný čas vydelil počtom iterácií (počet vkladania a mazania prvku – 100 000) a vyšiel priemerný čas pre operácie vkladania a mazania neoptimalizovaného a optimalizovaného stromu.

V tejto tabuľke môžeme vidieť časy pred a po optimalizácií.

|  |  |
| --- | --- |
| Operácia | Priemerný čas (ms) |
| Insert – neoptimalizovaný strom | 0,00243 |
| Delete – neoptimalizovaný strom | 1,85237 |
| Insert – optimalizovaný strom | 0,00852 |
| Delete – optimalizovaný strom | 0,00369 |

V tejto tabuľke môžeme vidieť parametre stromu pred a po optimalizácií.

|  |  |  |
| --- | --- | --- |
| Parameter | Stará hodnota | Nová hodnota |
| Minimum X | 0 | -2 326 660.15625 |
| Minimum Y | 0 | -2 326 660.15625 |
| Maximum X | 10 000 000 | 10 000 000 |
| Maximum Y | 10 000 000 | 10 000 000 |
| Výška | 5 | 15 |

Ako druhý test efektívnosti optimalizácie som sme si vytvorili strom o rozlohe 10 000 000 \* 10 000 000 a výške 5. Následne sme vložili do neho 500 000 prvkov do celého stromu a 500 000 prvkov ktoré sa vložili len do severo-západného syna koreňa. Následne sa vkladalo a mazalo 100 000 prvkov po celej rozlohe stromu, zmerala sa rýchlosť operácií Insert a Delete, vykonala sa optimalizácia stromu a znova sa vkladalo a mazalo 100 000 prvkov a merala sa rýchlosť operácií.

V tejto tabuľke môžeme vidieť časy pred a po optimalizácií.

|  |  |
| --- | --- |
| Operácia | Priemerný čas (ms) |
| Insert – neoptimalizovaný strom | 0,00217 |
| Delete – neoptimalizovaný strom | 0,48311 |
| Insert – optimalizovaný strom | 0,00917 |
| Delete – optimalizovaný strom | 0,00306 |

V tejto tabuľke môžeme vidieť parametre stromu pred a po optimalizácií.

|  |  |  |
| --- | --- | --- |
| Parameter | Stará hodnota | Nová hodnota |
| Minimum X | 0 | -1 250 000 |
| Minimum Y | 0 | -1 250 000 |
| Maximum X | 10 000 000 | 10 000 000 |
| Maximum Y | 10 000 000 | 10 000 000 |
| Výška | 5 | 14 |

Po vykonaní optimalizácie môžeme vidieť, že čas potrebný pre vykonanie metódy Insert sa zhoršil, z dôvodu zmeny výšky stromu, čiže sa musí dlhšie hľadať správny uzol do ktorého sa prvok vloží. Na opačnej strane, operácia Delete sa niekoľkonásobne zlepšila, prvky sa rovnomernejšie rozložili medzi uzly takže sa bude prechádzať menej prvkov pri hľadaní konkrétneho prvku určeného na vymazanie.

**Popis systému**

Systém bude obsahovať dva quad stromy typu nehnuteľnosť a parcela. Strom sa skladá z uzlov, ktoré zodpovedajú za danú oblasť. V tejto oblasti budú dane nehnuteľnosti/parcely. Pri vložení novej nehnuteľnosti sa pozrie do quad stromu parciel na súradnice vkladanej nehnuteľnosti a pridá jej referenciu do každej nájdenej parcely a nehnuteľnosti pridá referenciu každej nájdenej parcely na ktorej sa nachádza. To isté platí keď sa pridáva do systému nová parcela.

Obrázok, na ktorom je text, snímka obrazovky, diagram, písmo

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, diagram, písmo

Automaticky generovaný popis

**UML model**

Obrázok, na ktorom je text, diagram, rovnobežný, písmo

Automaticky generovaný popis

**Časové zložitosti operácií**

|  |  |
| --- | --- |
| Premenná | Popis |
| M | Počet nehnuteľností |
| N | Počet parciel |
| I | Počet parciel na ktorých sa nehnuteľnosť nachádza |
| J | Počet nehnuteľností ktoré sa nachádzajú na parcele |
| K | Počet parciel na ktorých sa nehnuteľnosť bude nachádzať po editácií |
| L | Počet nehnuteľností ktoré sa budú nachádzať na parcele po editácií |

|  |  |  |
| --- | --- | --- |
| Úloha | Priemerná zložitosť | Najhoršia zložitosť |
| 1. | O(Log M) | O(M) |
| 2. | O(Log N) | O(N) |
| 3. | O(Log M + Log N | O(M + N) |
| 4. | O(Log M + Log N + I) | O(M + N + I) |
| 5. | O(Log N + Log M + J) | O(N + M + J) |
| 6. | O(Log M + Log N + I + K) | O(M + N + I + K) |
| 7. | O(Log N + Log M + J + L) | O(N + M + J + L) |
| 8. | O(Log M + Log N + I) | O(M + N + I) |
| 9. | O(Log N + Log M + J) | O(N + M + J) |

**Užívateľská príručka**

Po zapnutí aplikácie má používateľ možnosť vytvoriť stromy so želanými parametrami a vygenerovať do nich zadaný počet prvkov o požadovanej rozlohe alebo načítať už existujúce stromy zo súboru. Pri vytváraní stromov si užívateľ zadá rozlohu, ktorú chce aby strom nadobúdal. Túto rozlohu zadá do text boxov X1, Y1, X2, Y2. Ďalej zadá výšku, akú chce aby stromy mali. Následne počet prvkov, ktoré sa budú generovať a ich šírku (X) a dĺžku (Y). Po vyplnení stlačí tlačidlo „Populate“ ktorý vytvorí stromy a vygeneruje do nich prvky. Následne môže používateľ vyhľadávať prvky na záložke „Find data“, kde má na výber vyhľadávanie parciel, nehnuteľností alebo oboch naraz. Po vyhľadaní môže používateľ prejsť na záložku „Edit data“ alebo „Delete data“ kde zadá index vypísaného prvku ktorý chce editovať alebo mazať. Pri editácií vyplní aj polia, ktoré chce zmeniť. Používateľ môže taktiež pridať prvok na záložke „Insert data“, kde si vytvorí vlastnú parcelu. V tomto prípade pozor na originalitu kľúča, ten je typu int a priradzuje sa prvkom počas generovania podľa toho, ktorý v poradí bol vygenerovaný, tak novým prvkom začnite dávať kľúče s hodnotou „počet generovaných prvkov v strome + 1“.