* Zložitosti
* Uml diagramy
* Popis vyhľadávania a mazania duplicít s ukážkami kódu

**K-D vyhľadávací binárny strom**

Prerekvizitou vkladania inštancií žiadaného typu do K-D stromu je implementácia rozhrania IkdComparable pre kľúč vkladaných dát. Typ kľúča tým deklaruje, že inštancia vkladaného typu je schopná byť porovnávaná s kľúčom už vložených dát v strome podľa špecifikovanej dimenzie. Ďalej sa na túto metódu porovnávania budeme referovať ako na „komparátor“. Určenie dimenzie znamená uvedenie celého čísla, ktoré môže nadobúdať hodnoty počnúc číslom 1 až po číslo celkovej najvyššej dimenzie kľúča. Napríklad, ak chcem ukladať do K-D stromu dáta s kľúčom, ktorý sa skladá z polohy X a polohy Y, pričom prvú dimenziu kľúča určuje poloha X a druhú dimenziu zas poloha Y, potom do metódy porovnania kľúčov môže byť pre dimenziu uvedené číslo z množiny {1,2}.

Operácie:

**Vloženie**

V K-D strome sa ukladajú dáta tak, že ich finálna pozícia sa určí postupným vetvením v strome – hľadaním miesta, kde uložiť dáta na základe hodnoty ich kľúča. V strome sa porovnávajú hodnoty kľúčov nasledovne:

1. Podľa aktuálnej úrovne porovnávaného prvku v strome sa určí dimenzia
2. Porovnajú sa všetky atribúty kľúča, ktoré určujú hodnotu aktuálnej dimenzie (zistenej v predcházajúcom bode) pre inštanciu dát, ktoré sa majú vložiť, s inštanciu aktuálne porovnávaných dát v strome pomocou komparátora
   1. Ak komparátor vráti výsledok, že vkladaný prvok s kľúčom má pre aktuálne porovnávanú dimenziu hodnotu kľúča menšiu alebo rovnú voči hodnote pre danú dimenziu prvku zo stromu, postupujeme v stromovej štruktúre smerom na ľavého syna.
   2. Pokiaľ je hodnota vkladaného kľúča v danej dimenzií väčšia ako existujúca v strome, postupuje sa smerom do pravého podstromu.
3. Tento postup sa aplikuje, kým porovnávaný prvok zo stromovej štruktúry nie je listom (nemá ani pravého, ani ľavého syna). Následne sa vloží prvok s kľúčom ako pravý alebo ľavý syn (podľa návratovej hodnoty komparátora) existujúceho listu.

Zložitosť operácie vkladania je pre n prvkov (bez degradácie stromu na zreťazený zoznam) logaritmická log2(n), pretože každým vetvením nemusíme porovnávať pravý alebo ľavý podstrom vďaka pravidlu, že hodnoty kľúča pre relevantnú dimenziu prvku v strome, ktoré sú menšie alebo rovné, budú vždy v ľavom podstrome.

**Vyhľadávanie**

* + riešenie vyhľadávania duplicít
  + unikátny prvok – nájdenie unikátneho prvku v k-d strome so sekundárnymi kľúčmi dosiahneme pridaním obmedzenia pre typ vkladaných dát tak, že musia realizovať rozhranie „ISame“, ktoré obsahuje metódu „isSame“. Na túto metódu sa ďalej budeme referencovať ako „Overovateľ“. Typ vkladaných dát sa realizáciou zaväzuje, že vie vyhodnotiť na základe unikátneho atribútu či je inštancia tá istá na základe referencie na inštanciu alebo hodnoty unikátneho atribútu. So splnením tejto prerekvizity sa v k-d strome vyhľadajú všetky duplicity na základe sekundárneho kľúča a následna sa z týchto duplicít nájde konkrétny prvok použitím overovateľa.
* Editovanie
* Vymazanie