AVL strom

1. Úvod
2. Stromy
   1. Obrázok, na ktorom je text

      Automaticky generovaný popisSpoločné prvky
      1. Trieda Node:

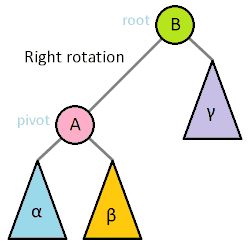
Tento útržok kódu predstavuje triedu Node (Uzol) Ktorá má 5 atribútov: key, value, height, left a right. Trieda tiež obsahuje konštruktor ktorý nastaví inicializačné hodnoty na tie ktoré sú zadané pri vytváraní objektu.

Obrázok, na ktorom je text

Automaticky generovaný popis

* 1. Rotácie

Rotácie menia root stromu tak že zvolia jedno z detí za root a správne priradia ostatné deti.



Delia sa na pravú rotáciu (RR) a ľavú rotáciu (LR)

* + 1. Obrázok, na ktorom je text

       Automaticky generovaný popisRR

Pri RR sa novým rootom stromu stane pri obrázku vyššie stane uzol A a ľavím dieťatom B sa stane pravé dieťa A.

* + 1. LR

Je opačná rotácia.

* + 1. Metóda minnode

Nájde najmenší prvok v strome

* 1. AVL strom
     1. Inicializacia:Obrázok, na ktorom je text

        Automaticky generovaný popis

V tejto časti kódu si vytvoríme triedu AVL a inicializujeme ju s atribútom root ktorý je neskôr v konštruktore nastvený na null a polom čísel ktoré bude neskôr použité na testovanie.

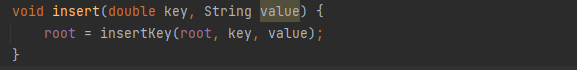
* + 1. Obrázok, na ktorom je text

       Automaticky generovaný popisMetóda getheight vráti hodnotu výšky uzlu ak uzol nie je null ak áno vráti 0
    2. Metóda updateheight aktualizuje výšku tak, že pripočíta k maximu výšok pravého a ľavého podstromu 1.
    3. Aktualizácia balance faktoru.   
       Pri strome typu AVL sa balancovanie stromu deje pomocou balance faktoru ktorý sa môže pohybovať len od -1 do 1. Balance faktor sa vypočíta odčítaním výšky ľavého podstromu od výšky pravého podstromu. Ak je výsledok menší ako -1 znamená to, že strom je hlbší v pravom podstrome. Zistíme či pravé dieťa pravého dieťaťa rootu ma menšiu hĺbku ako ľavé dieťa pravého dieťaťa rootu. Ak áno potrebujeme spraviť jednu pravú rotáciu na pravom dieťati rootu, aby sme dostali strom ktorého pravé dieťa je prevážené doprava. Po tomto kroku spravíme ľavú rotáciu na roote a dostaneme vybalancovaný strom.

Obrázok, na ktorom je text

Automaticky generovaný popis

* + 1. Insert

Metóda insert vkladá do rootu kľúč a údaj aby nebola potreba passovať do funkcie root, metóda je rozdelená na dve.

* + 1. Obrázok, na ktorom je text

       Automaticky generovaný popisInsertKey  
       Používa klasickú rekurzívnu BST insert metódu na vkladanie s rozdielom, že na konci je použitá metóda update height a update balance.
    2. Hľadanie v strome

Pri hľadaní v AVL strome je použitá klasická rekurzívna BST vyhľadávacia metódaObrázok, na ktorom je text

Automaticky generovaný popis

Obrázok, na ktorom je text

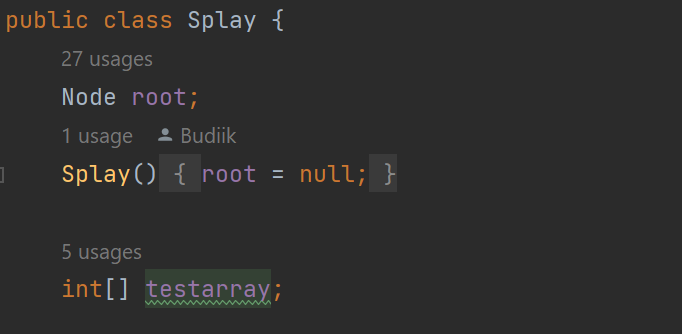
Automaticky generovaný popis

* + 1. Metóda delete a deleteNode

Vymazávanie je robené podobne ako insert klasickou rekurzívnou BST vymazávanou metódou, s rozdielom, že na konci je použitá metóda update height a update balance.

* 1. Splay strom

Splay strom, je samovyvažujúci sa binárny vyhľadávací strom. Splay strom umožňuje efektívny prístup k nedávno používaným prvkom tým, že po každej operácii (vkladaní, vyhľadávaní, odstraňovaní) sa vyvažuje pomocou špeciálnych rotácií tak aby sa daný prvok stal rootom stromu.

* + 1. Inicializácia

V tejto časti kódu si vytvoríme triedu Splay a inicializujeme ju s atribútom root ktorý je neskôr v konštruktore nastvený na null a polom čísel ktoré bude neskôr použité na testovanie.

* + 1. Insert

Obrázok, na ktorom je text

Automaticky generovaný popisPodobne ako v AVL strome sme použili dve metódy, kvôli jednoduchosti používania. Začíname s klasickým rekurzívnym bst vkladaním no po tom čo nájdeme správne miesto a vložíme doň nový uzol. Sa pri každom rekurzívnom kroku pozrieme či jeden z jeho pradetí nie je uzol ktorý sme práve pridali ak je vykonávame správnu splay rotáciu. Ak je daný uzol práve dieťa pravého dieťaťa spravíme 2 ľavé rotácie čím posunieme dané dieťa na vrch ak je daný uzol ľavé dieťa práve ho dieťaťa spravíme 1 pravú rotáciu na pravom dieťati a 1 ľavo na roote ak je daný uzol 1 z detí ľavého dieťaťa opakujeme opačné rotácie. Po skončení druhej funkcie sa vykoná posledná rotácia ak je uzol pravé alebo ľavé dieťa rootu.

Obrázok, na ktorom je text

Automaticky generovaný popis

* + 1. Search

Metóda search je podobná k metóde insert, s rozdielom že sa nevkladá nový uzol.

1. Testovanie
   * 1. Testovací postup

Na testovanie som použil metódu ktorá najprv vytvorí pole o veľkosti Testing\_size (zadaná používateľom v testovacom súbore), ktoré má v náhodnom poradí všetky čísla od 0 do Testing\_size (toto zaručí funkcia generateUniqueArray). Následne meriame čas ktorý je treba na vloženie Test\_jump prvkov a zapisujeme ich do súboru.

Takto postupujeme pre všetky funkcie.

Obrázok, na ktorom je text

Automaticky generovaný popis

* 1. Výsledky testovania

Výsledky som vizualizoval v jazyku python pomocou knižnice matplotlib.

Obrázok, na ktorom je text

Automaticky generovaný popis

Obrázok, na ktorom je tabuľka

Automaticky generovaný popis