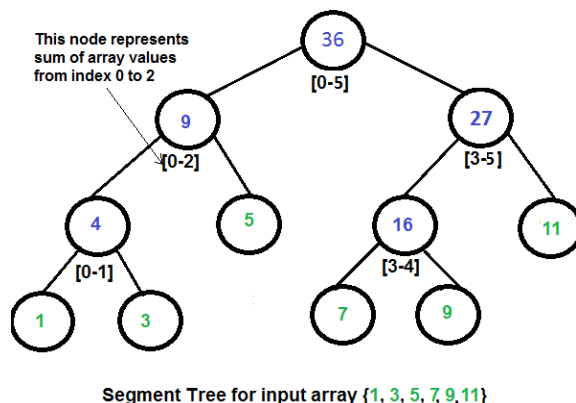


OBJEKTNO ORIJENTISANO PROGRAMIRANJE

- domaći zadatak broj 2 -

Funkcionalna specifikacija

Na programskom jeziku C++ implementirati biblioteku klasa za dinamičku reprezentaciju stabla intervalnog pretraživanja (*RSearchTree*) i preklopiti operatore za klasu *BigDecimal*. Stablo se izgrađuje na osnovu nesortiranog niza velikih decimalnih brojeva. U nastavku teksta opisane su funkcionalnosti za klase *RSearchTree* i *TreeNode* i *BigDecimal*.



Slika 1 Primer stabla intervalnog pretraživanja dobijenog od niza brojeva [1,3,5,7,9,11].

Specifikacija funkcionalnosti klase *TreeNode*.

Čvor stabla sadrži početni indeks intervala (*start*), završni indeks intervala (*end*), vrednost (*value:BigDecimal*), kao i pokazivače na levo (*left*) i desno (*right*) podstablo. Vrednost je dinamički alociran decimalni broj.

Stvaranje čvora.

Implementirati konstruktor `TreeNode(int start, int end, const BigDecimal* val)`.

Čvor se može napraviti kao kopija drugog čvora.

Implementirati konstruktor kopije koji kopira samo indekse i vrednost (pokazivači kopije su `nullptr`).

Dohvatanje/postavljanje levog/desnog podstabla.

Metoda vraća adresu odgovarajućeg podstabla.

```
TreeNode* getLeft(); i TreeNode* getRight ();
```

Metoda postavlja levo/desno podstablo čiji koren je zadat kao argument.

```
void setLeft (TreeNode*); i void setRight (TreeNode*);.
```

Dohvatanje početnog/završnog indeksa.

```
int getStart(); i int getEnd();
```

Da li čvor pokriva interval?

Implementirati metodu `bool covers(int start, int end)`; koja proverava da li čvor pokriva zadati interval. Vraća `true` ukoliko je zadati interval identičan ili obuhvata interval čvora.

Ispis čvora stabla.

Preklopiti operator `<<` koji ispisuje čvor u formatu *vrednost [start: end]*.

Obezbediti propisno brisanje čvora.

Implementirati destruktora koji prilikom brisanja uklanja dinamički alociranu vrednost.

Specifikacija klase *RSearchTree*

Dinamičko stablo intervalnog pretraživanja je binarno stablo intervala koje se pravi na osnovu nesortiranog niza brojeva na način koji je opisan u nastavku. Stablo sadrži čvorove klase *TreeNode*. Klasa stabla ima privatno polje `root` koje predstavlja pokazivač na koreni čvor stabla.

Stablo se stvara na osnovu zadatog ulaznog niza .

Implementirati konstruktor klase `RSearchTree(vector<BigDecimal*>& values)`; . U svakom koraku trenutni podniz (interval) deli se na dve polovine. Od leve polovine izgrađuje se levo

podstablo, a od desne polovine desno podstablo. Ukoliko interval koji se deli ima neparan broj elemenata, središnji element pripada levoj polovini intervala. Ceo niz je početni interval. Postupak se završava kada se deobom dođe do prostih intervala sa jednim elementom. Svi elementi niza na kraju pravljenja stabla moraju da budu u listovima stabla (videti primer). Interni čvorovi stabla sadrže zbir listova u svom podstablu. Vrednost u listu je broj niza. Ne sme se koristiti rekurzija – implementirati iterativno rešenje.

Stablo se može stvoriti kao kopija već postojećeg stabla.

Implementirati obe varijante konstruktora kopije (duboko kopiranje i kopiranje sa premeštanjem) za potrebe kloniranja stabala. Ne sme se koristiti rekurzija – implementirati iterativno rešenje.

Stablu se može dodeliti drugo stablo.

Preklopiti obe varijante operatora za dodelu vrednosti (operator=), sa kopiranjem i sa premeštanjem. Ne sme se koristiti rekurzija – implementirati iterativno rešenje.

Minimalni broj čvorova koji pokrivaju zadati interval.

Implementirati javnu metodu `vector<TreeNode*> intervalNodes(int start, int end);`. Privatna metoda vraća minimalni broj čvorova kojima je obuhvaćen zadati interval. Na primer, za stablo u primeru `intervalNodes(2, 4)` vraća niz čvorova sa vrednostima 5 i 16. Ne sme se koristiti rekurzija – implementirati iterativno rešenje.

Zbir elemenata u podnizu.

Implementirati javnu metodu `BigDecimal intervalSum(int start, int end);`. Metoda računa zbir elemenata podniza određenog početnih indeksom *start*, i završnim indeksom *end*. Interval je inkluzivan. Ukoliko zadati interval ne obuhvata indekse niza, rezultat je 0. Implementaciju svesti na poziv metode *intervalNodes*. Na primer, *intervalSum(2,4)* daje rezultat 21. Ne sme se koristiti rekurzija – implementirati iterativno rešenje.

Ispisivanje stabla.

Implementirati ispis čvorova stabla obilaskom po širini. Za dato stablo ispis treba da izgleda na sledeći način: 36[0:5],9[0:2],27[3:5],4[0:1],5[2:2],16[3:4],11[5:5],1[0:0],3[1:1],7[3:3],9[4:4]. Preklopiti operator<< koji poziva navedenu metodu za ispis.

Omogućiti propisno uništavanje stabla.

Implementirati destruktora koji treba da oslobodi sve čvorove stabla.

Specifikacija dopuna klase BigDecimal

Preklopiti sledeće operator za klasu iz prvog domaćeg zadatka, `BigDecimal`: +, -, ==, !=, <, >, <=, >=, unarni -, unarni+ (aps. vrednost) << (pomeranje tačke levo), >> (pomeranje tačke desno), >>= (uklanjanje tačke).

Test funkcija

U projektu glavnog programa treba da postoji funkcija `void test();` koja testira rad sa velikim brojevima i stablom. Studenti treba da implementiraju datu funkciju i uslovno je prevode ako nije definisan makro `PROF_TEST`. Funkcija `main` treba samo da pozove `test` funkciju i na kraju ispiše njeno trajanje u milisekundama, korišćenjem tipova i operacija iz zaglavlja `<ctime>`. Test funkcija mora biti prevedena sa izvornim kodom rešenja.

Tehnički zahtevi i smernice za izradu rešenja

Sve klase i metode moraju biti imenovane prema zahtevima iz domaćeg zadatka. Svaka klasa koja koristi dinamičku memoriju mora biti bezbedna za korišćenje. Programski kod klasa rasporediti u odgovarajuće **.h** i **.cpp** fajlove. Nije dozvoljeno korišćenje globalnih promenljivih za razmenu podataka. Ukoliko u zadatku nešto nije dovoljno jasno definisano, treba usvojiti razumnu pretpostavku i na temeljima te pretpostavke nastaviti izgrađivanje svog rešenja.

VAŽNE NAPOMENE

Za uspešno odbranjen domaći zadatak potrebno je na odbrani pokazati kod podeljen na odgovarajuće projekte, .h i .cpp fajlove.

- Klase kojima su implementirani osnovni koncepti treba da budu smeštene u poseban projekat rešenja koji se prevodi kao statička biblioteka (`BigDecimal.lib`).
- Klase koje implementiraju stablo treba da budu smeštene u poseban projekat rešenja koji se prevodi kao statička biblioteka (`tree.lib`).
- Glavni program napisati u posebnom projektu koji se prevodi kao Win32 Console Application (`dz2_test.exe`) i povezati sa bibliotekama za stabla i decimalne brojeve.
- NIJE DOZVOLJENO SMESTITI CEO KOD U JEDAN PROJEKAT ILI CPP fajl!
- Na kraju programa potrebno je ispisati na standardnom izlazu vreme trajanja funkcije `test` u milisekundama. Može se iskoristiti kod za merenje vremena na jeziku C++ koji je dat u zadatku 2.8 u materijalima za vežbe.

26.11.2017. godine

sa predmeta