

Fajlovi "lista.h" i "lista.cpp" su fajlovi koje smo izgradili na vježbama.

U fajlu "main.cpp" sam istestirao sve konstruktore, operatore dodjela, te sve friend operatore i operatore koji mijenjaju instancu klase Matrica, kao i operator ispisa.

U fajlu "matrica.h" napravljena je klasa Matrica sa odgovarajućim atributima (koji su privatni).

Ti atributi su:

- 1) int broj_redova - stvarni broj redova u matrice,
- 2) int broj_kolona - stvarni broj kolona u matrice,
- 3) Lista<pair<int, Lista<pair<int, double>>>> M;

npr. Neka je dato M: $\{\{0, \{\{0,1\}, \{1,2\}\}\}, \{11, \{\{11,12\}, \{12,13\}\}\}\}$ sa 100 redova i 100 kolona. Tada imamo matricu:

- a) u kojoj redovi 0 i 11 imaju nenula elemente,
- b) u redu 0: u koloni 0 je element 1, u koloni 1 je element 2,
- c) u redu 11: u koloni 11 je element 12, u koloni 12 je element 13,
- d) sve pozicije izuzev navedene 4 ne pamtim u M jer mi one predstavljaju nule.

Opisi implementacija:

- 1) init
- 2) konstruktor kopije
- 3) operator dodjele
- 4) destruktor
- 5) pomjerajući konstruktor kopije
- 6) pomjerajući operator dodjele
- 7) konstruktor sa dva parametra
- 8) funkcija kriterij

-treba mi da bi sortirao `vector<pair<pair<int, int>, double>>` u konstruktoru ispod. Sortira ih po rednom broju reda i rednom broju kolone.

9) konstruktor sa 3 parametra:

- a) `const int &broj_redova,`
- b) `const int &broj_kolona,`
- c) `vector<pair<pair<int, int>, double>>` elementi.

npr. Neka su broj redova 100 i broj kolona 100 i elementi:
{{{2,10},5},{{{100,50},25},{{{2,400},15},{{{700,800},35}}}} proslijeđeni ovom konstruktoru.

Tada će konstruktor uraditi sljedeće:

- pozvati init,

- sortirati pomenutu stukturu po 8) (ona će sada izgledati ovako:
{{{2,10},5},{{{2,400},15},{{{100,50},25},{{{700,800},35}}}},

- napraviti Lista<pair<int,double>> red (prvi član para predstavlja redni broj kolone elementa koji je drugi član para),

- dodati prvi element iz vektora elementi u listu red, naravno dodavajući samo, kao par, broj kolone i vrijednost,

- u for petlji će se, ukoliko se radi o elementima u istom redu, na kraj liste red dodavati, kao par, broj kolone i vrijednost i-tog elementa,

- čim i-ti element ne bude u istom redu kao (i-1). element, u listu M dodat ćemo, kao par, broj reda (i-1). elementa i listu red,

- očistiti listu red i dodati u red, koji je sada prazan, broj kolone i vrijednost i-tog elementa iz vektora elementi (kao par),

- izvršiti krajnje dodavanje u listu M.

10) funkcija getElement

Ima 2 parametra:

- a) const int &i - broj reda

- b) const int &j - broj kolone

Vraća element na poziciji (i,j), međutim vraća 0 ukoliko ne postoji pozicija (i,j).

11) operator+

Ima 2 parametra:

- a) const Matrica &A,

- b) const Matrica &B.

Radi sljedeće:

- baca izuzetak ako dimenzije matrica A i B nisu iste,

- pravi matricu za_vracanje koju inicijalizira sa matricom A (dakle pozove se odgovarajući konstruktor kopije),

- prolazi kroz listu M matrice B, te za svaki element te liste M prolazi kroz listu M matrice za_vracanje, u ovoj for petlji traži i-ti red iz matrice B u matrici za_vracanje, ako ga nema, onda na odgovarajuću

poziciju u listu M matrice za_vracanje insertuje i-ti red iz liste M matrice B. Ako pronađe i-ti red iz liste M matrice B u listi M matrice za_vracanje, onda sabira redove matrice B i matrice za_vracanje na odgovarajući način. Taj odgovarajući način je: prolazi kroz sve elemente u i-tom redu liste M matrice B, u toj svakoj iteraciji prolazi kroz odgovarajuće elemente u i-tom redu liste M matrice za_vracanje i traži odgovarajuću kolonu. Ukoliko ta kolona postoji, ako odgovarajući elementi sabrani daju 0 onda tu poziciju briše (jer ne pamtim nule u matrici), u suprotnom povećava element na toj poziciji za element iste pozicije iz liste M matrice B. Međutim ukoliko ta kolona ne postoji, onda na odgovarajuće mjesto insertuje element iz liste M matrice B, odnosno push_back-a ako se ne izvrši insert,

-na kraju vrati matricu za_vracanje.

12) operator*

Ima 2 parametra:

- a) const double &broj,
- b) const Matrica &A.

Radi sljedeće:

-ako množi nulom samo vrati praznu matricu odgovarajućih dimenzija,

-u suprotnom napravi matricu za_vracanje koju inicijalizuje sa matricom A, te svaki element liste M matrice za_vracanje mijenja na način da ga pomnoži sa brojem broj,

-na kraju vrati matricu za_vracanje.

13) operator*

Ima 2 parametra:

- a) const Matrica &A,
- b) const double &broj.

Radi sljedeće:

-odradi sve preko 12).

14) operator-

Ima 2 parametra:

- a) const Matrica &A,
- b) const Matrica &B.

Radi sljedeće:

-baca izuzetak ako dimenzije matrica A i B nisu iste,

-odradi sve preko 11) i 12).

15) operator*

Ima 2 parametra:

- a) const Matrica &A,
- b) const Matrica &B.

Radi sljedeće:

- baca izuzetak ako dimenzije matrica A i B nisu odgovarajuće da bi se moglo izvesti množenje istih,
- pravi praznu matricu za vraćanje odgovarajućih dimenzija,
- prolazi for petljom kroz redove liste M matrice A, pravi varijablu bool dodajRed i setuje na false, te pravi pair<int, Lista<pair<int, double>>> red, (i)
 - za svaki red iznad prolazi kroz sve tehnički moguće kolone matrice B, tj. od 0 do B.broj_kolona, te svakom iteracijom pravi double vrijednost i setuje je na 0, (j)
 - za svaku kolonu iznad prolazi sve elemente u i-tom redu liste M matrice A, te povećava vrijednost na odgovarajući način,
 - nakon završetka zadnje for petlje, tj. nakon prolaska kroz sve elemente u i-tom redu liste M matrice A, ako vrijednost nije 0 onda u red red dod kao par j i vrijednost, te setuje varijablu dodajRed na true,
 - nakon završavanja srednje for petlje, ako je dodajRed true u listu M matrice za vraćanje push_back-a red red,
 - na kraju vrati matricu za vraćanje.

16) operator^ (odrađen rekursivno)

Ima 2 parametra:

- a) const Matrica &A,
- b) const int &stepen.

Radi sljedeće:

- baca izuzetak ako A nije kvadratna matrica,
- baca izuzetak ako je stepen negativan,
- ako je stepen 0 onda pravi jediničnu matricu odgovarajućih dimenzija i vraća je,
- ako je stepen 1 vraća A,
- ako je stepen paran vraća tehnički $(A^2)^{(\text{stepen}/2)}$, što je isto što i A^{stepen} ,
- ako je stepen neparan vraća tehnički $(A^2)^{(\text{stepen}-1)/2} * A$, što je isto što i $A^{(\text{stepen}-1)} * A$, a što je ustvari opet A^{stepen} .

Dakle svodim problem na manje potprobleme

17) operator-

Ima 1 parametar:

a) const Matrica &A.

Radi sljedeće:

-odradi sve preko 12).

18) operator* (operator transponovanja)

Ima 1 parametar:

a) const Matrica &A.

Radi sljedeće:

-na pametan način koristi konstruktor sa tri parametra.

19) operator+= (mijenja matricu nad kojom se poziva dodavajući matricu koja je parametar)

Ima 1 parametar:

a) const Matrica &A.

Radi sljedeće:

-odradi sve preko 11).

20) operator*= (mijenja matricu nad kojom se poziva množeći je sa brojem koji je parametar)

Ima 1 parametar:

a) const double &broj.

Radi sljedeće:

-odradi sve preko 13).

21) operator-= (mijenja matricu nad kojom se poziva oduzimajući od nje matricu koja je parametar)

Ima 1 parametar:

a) const Matrica &A.

Radi sljedeće:

-odradi sve preko 14).

22) operator*= (mijenja matricu nad kojom se poziva množeći je sa matricom koja je parametar)

Ima 1 parametar:

a) const Matrica &A.

Radi sljedeće:

-odradi sve preko 15).

23) operator^= (mijenja matricu nad kojom se poziva stepenujući je sa brojem koji je parametar)

Ima 1 parametar:

a) const int &broj.

Radi sljedeće:

-odradi sve preko 16).

24) operator<< za const Matrica &A

Radi sljedeće:

-ispisuje matricu A u formatu:

broj_redova broj_kolona

(i,j):e (gdje su: i-red, j-kolona, e-vrijednost na poziciji (i,j))

-nakon ispisivanja svih elemenata jednog reda, ispis prelazi u novi red.