Rīgas Tehniska universitāte

Datorzinātnes un informācijas tehnoloģiju fakultāte

Lietišķo datorsistēmu institūts

2. kurss grupa DB 9

Studiju darbs mācību priekšmetā

„Diskrētas struktūras datorzinātnēs”

Izstrādāja : Sergejs Terentjevs

Nodošanas datums : \_\_.\_\_.\_\_

Pārbaudīja : Judīte Ciekure

2007/2008 mācību gads.

# Anotācija

Studiju darba pārskatā ir aprakstīts grafs, dota definīcija grafa jēdzienam un tā viediem, izstrādāts un pamatots teorētiskais pamatojums:

1. Grafa Blakus virsotņu un incidences matricai, to virsotņu lokālajam pakāpēm.
2. Prefiksa un postfiksa pierakstām, vispārīga veida doti pirmssakārtojuma un pēcsakārtojuma algoritmu darbības principi.

Pārskata ir parādīti paskaidrojumi programmas lietotājam. Studiju darbā tika izstrādātas un atkļūdotas divas programmas Microsoft Visual Studio.Net 2005 vidē, Visual Basic 2005 programmēšanas valodā. Studiju darba pārskata apjoms satur: 26. lpp izklāsta teksta, 27. attēlus, literatūras sarakstu un pielikumu.

# Saturs

[Anotācija 2](#_Toc187458404)

[Saturs 3](#_Toc187458405)

[1. Uzdevuma Nostādne 4](#_Toc187458406)

[1.1. Uzdevuma sākotnējais teksts 4](#_Toc187458407)

[2. Teorētiskais pamatojums 5](#_Toc187458408)

[2.1. 1 .uzdevumam 5](#_Toc187458409)

[2.2. 2 .uzdevumam 9](#_Toc187458410)

[3. Paskaidrojumi programmas lietotājam 12](#_Toc187458411)

[3.1. 1.uzdevumam 12](#_Toc187458412)

[3.1.1. Darba uzsākšana 12](#_Toc187458413)

[3.1.2. Blakus virsotnes un incidences matricas formas 15](#_Toc187458414)

[3.2. 2.uzdevumam 17](#_Toc187458415)

[4. Kontroles piemēra analīze 19](#_Toc187458416)

[4.1. 1.uzdevumam 19](#_Toc187458417)

[4.2. 2.uzdevumam 23](#_Toc187458418)

[Secinājumi 25](#_Toc187458419)

[Literatūras saraksts 26](#_Toc187458420)

# Uzdevuma Nostādne

## Uzdevuma sākotnējais teksts

Varianta numurs ir 9. Variants ietver trīs uzdevumus, no kuriem divi ir obligāti.

Varianta prasības:

* Katra uzdevuma atrisināšanai jāsastāda programma;
* Programma veic datu ievadi prasītājā formā;
* Programma atrisina uzdevumu;
* Programma izvada rezultātu;
* Jānodod programmas .EXE fails.

1. Iegūt grafa blakus virsotņu matricu, incidences matricu, noteikt virsotņu lokālo pakāpi. Doto neorientēto grafu ievada ar loku sarakstu (virsotņu pāri, kas nosaka, virsotnes apzīmē ar burtiem), grafa maksimālais virsotņu skaits 15, maksimālais loku skaits 20.
2. Izteiksmju vērtības aprēķināšana. Tiek ievadīta izteiksme, noteikt tās veidu (prefiksa/postfiksa ) un aprēķināt vērtību. Izmantotie simboli: „ **+** ”, „ **-** ”, „ **\*** ” , „ **^** ”, „ **/** ”. Jābūt iespējai apstrādāt divciparu skaitļus un izpildīt vismaz 8 darbības.

# Teorētiskais pamatojums

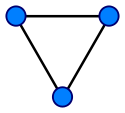
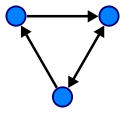
## 1 .uzdevumam

Par grafu G sauc sakārtotu kopu pāri G = (V, E), kur V ir galīga un netukša kopa, bet E sastāv no kopas V elementiem un pāriem, t.i., . Kopas V elementus sauc par grafa virsotnēm, savukārt, kopas E (a, b) elementus sauc grafa šķautnēm jeb lokiem, šeit a un b ir loka E galapunkti. Tātad loks E savieno grafa virsotnes a un b.

Atzīmēsim, ka dažas no kopām E (a, b) drīkst būt tukšas, tātad grafa loku var vispār nebūt, tādu grafu sauc par tukšo grafu .

Ja E=(a, b) ir nesakārtoti virsotņu pāri, tad grafu sauc par neorientētu (att. 2.1). Šī grafa kāda no šķautnēm var vest gan no a virsotnes uz b virsotni, gan arī no b virsotnes uz a virsotni.

Savukārt, ja E = (a, b) ir sakārtoti pāri, grafs ir orientēts (att. 2.2). Orientētā grafā lokus attēlo ar bultām ( jo kārtība ir svarīga).

[](http://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5:Undirected.svg) [](http://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5:Directed.svg)

2.1 att. Neorientēts grafs. 2.2 att. Orientēts grafs.

Katram grafam G = (V, E) atbilst noteikta attieksme, ko sauc par incidenci. Saka, ka divas grafa virsotnes ir incidentas, ja tās ir savienotas ar loku. Ņemot vērā incidenci, katram grafam var piekārtot tā blakus virsotņu un incidences matricu. Gan blakus virsotņu, gan incidences matrica viennozīmīgi nosaka grafu.

Blakus virsotņu matrica ir kvadrātveida matrica, kur V ir grafa virsotņu kopa. Neorientētam grafam blakus virsotņu matricas elements  ir vienāds ar 1, ja grafa *i*-tā virsotne ir incidenta ar *j*-to virsotni; un vienāds ar 0, ja tā nav.

A (i, j) = 1, ja loks ( i, j ) ∈ E

A (i, j) = 0, ja loks ( i, j ) ∉ E

Neorientēta grafa blakus virsotņu matrica ir simetriska attiecība pret galveno diagonāli. Savukārt, galvenās diagonāles  elementa vērtība ir 0, jo loka E virsotņu pārī virsotne nevar atkārtoties, loks matricā nevar būt par cilpu.

Piemērs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 1 | 0 | 1 | 1 |
| B | 1 | 0 | 1 | 0 | 0 |
| C | 0 | 1 | 0 | 0 | 0 |
| D | 1 | 0 | 0 | 0 | 1 |
| E | 1 | 0 | 0 | 1 | 0 |

Dotajam grafam ir 5 virsotnes, tātad grafa blakus virsotņu matrica ir 5×5. Blakus virsotņu matrica ir simetriska.

A (V **i** , V **j**) =

Orientēta grafa blakus virsotņu matricas elements  ir vienāds ar 1, ja no grafa *i*-tās virsotnes iet šķautne uz *j*-to virsotni; pretējā gadījumā šīs elements vienāds ar 0.

A (i, j) = 1, ja loks < i, j > ∈ E

A (i, j) = 0, ja loks < i, j > ∉ E

Orientēta grafa blakus virsotņu matrica ir asimetriska attiecība pret galveno diagonāli.

Piemērs:

Dotā grafa blakus virsotņu matrica ir 6×6 matrica, jo grafam ir 6 virsotnes. Grafa rindā elements 1 atbilst izejošam lokam.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A | 0 | 1 | 0 | 1 | 0 | 1 |
| B | 0 | 0 | 1 | 0 | 0 | 0 |
| C | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 1 | 1 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 1 | 0 |

A (V **i** , V **j**) =

Incidences matrica ir taisnstūrveida matrica , kur rindas atbilst grafa virsotnēm, kolonas atbilst grafa lokiem. Neorientēta grafa incidences matricas katra kolonna satur divus 1, pārējas ir 0. Ja *j*-tais loks savieno *i*-to un *l*-to virsotnes, tad *j*-tajā kolonnā *i*-tajā un *l*-tajā rindās ir 1, bet pārējie elementi šajā kolonnā ir 0.

B (i, j) **=** 1, ja virsotne Vi incindenta lokam E j

B (i, j) **=** 0, ja virsotne Vi nav incindenta lokam E j

Piemērs:

*E*2

*E*1

*E*3

*E*5

*E*4

*E*6

Dotajam grafam ir 6 loki, tātad grafa incidences matricai ir 5 rindas un 6 kolonnas.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *E*1 | *E*2 | *E*3 | *E*4 | *E*5 | *E*6 |
| A | 1 | 1 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 1 | 0 | 0 |
| C | 0 | 1 | 0 | 0 | 1 | 0 |
| D | 0 | 0 | 0 | 1 | 1 | 1 |
| E | 1 | 0 | 1 | 0 | 0 | 1 |

B (V **i** , V **j**) =

Orientēta grafa incidences matricas katrā kolonnā ir viens 1, viens -1, pārējie elementi ir 0. *j*-tās kolonnas *i*-tajā rindā ir 1, *l*-tajā rindā ir –1, ja *j*-tā loks iet no *i*-tās virsotnes uz *l*-to.

B (i, j) **=** +1, ja V i incidenta lokam E j un E j iziet no V i

B (i, j) **=** –1, ja V i incidenta lokam E j un E j ieiet V i

B (i, j) **=** 0, ja V i nav incidenta lokam E i

Piemērs:

*E*1

*E*3

*E*5

*E*6

*E*7

*E*4

*E*2

Dotā grafa incidences matrica satur 6 rindas un 7 kolonnas. Kolonnā elements 1 atbilst izejošajai šķautnei,  - ieejošam lokam.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *E*1 | *E*2 | *E*3 | *E*4 | *E*5 | *E*6 | *E*7 |
| A | +1 | +1 | +1 | 0 | 0 | 0 | 0 |
| B | 0 | - 1 | 0 | +1 | 0 | - 1 | 0 |
| C | 0 | 0 | 0 | - 1 | - 1 | 0 | 0 |
| D | 0 | 0 | - 1 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | +1 | +1 | - 1 |
| F | - 1 | 0 | 0 | 0 | 0 | 0 | +1 |

B (V **i** , V **j**) =

Par virsotnes lokālo pakāpi sauc loku skaitu, kas ir incidenti virsotnei (šī virsotne ir loku galapunkts).

Neorientēta grafa kādas virsotnes lokāla pakāpe ir loku skaits, kas ieiet šajā noteiktajā virsotne, piemēram, dots sekojošs grafs:

Grafa virsotņuV1, V2, V3 un V4 lokālas pakāpes ir 2.

Orientētam grafa virsotnēm eksistē izejošas un ieejošas lokālas pakāpes. Lokālo pakāpi izejošiem lokiem apzīmē ar dG+, bet ieejošiem lokiem apzīmē ar dG- .

Dotajām grafam virsotņu izejošas un ieejošas lokālas pakāpes ir :

* V1 virsotnei dG+(V1) = 2 un dG-(V1) = 1,
* V2 virsotnei dG+(V2) = 1 un dG-(V2) = 1,
* V3 virsotnei dG+(V3) = 0 un dG-(V3) = 1,
* V4 virsotnei dG+(V4) = 2 un dG-(V4) = 1.

## 2 .uzdevumam

Grafu teorija koks ir saistīts neorientēts grafs, kas nesatur ciklus. Sakārtotam kokam eksistē trīs apiešanas algoritmi, kas ļauj sistemātiski apmeklēt visas koka virsotnes sākot ar sakni. Pastāv šādi apiešanas algoritmi:

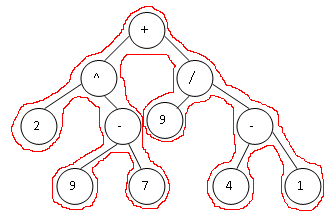
* Pirmssakārtojuma algoritms – pirms sakārtojuma apiešana sākas ar saknes apmeklējumu, tad tiek apmeklēts pirmais apakškoks, tad otrais u.t.t., līdz tiek apmeklēts viss koks.
* Iekšēja sakārtojuma apiešanas algoritms – realizējot iekšējā sakārtojuma apiešanu sakārtojuma rindā kā pirmo ievieto pirmo saknes pēcteci, tad sakni un tad visus pārējos tās pēctečus atrašanas kārtība.
* Pēcsakārtojuma algoritms - realizējot pēcsakārtojuma apiešanu sakārtojuma rindā pirmos ievieto saknes pēctečus un tikai tad pašu sakni.

Pielietojot minētos algoritmus, ar bināra koka palīdzību var attēlot izteiksmes, tad koku jāsāk būvēt no apakšas - tā strupceļa virsotnes ir simboli (operandi), bet iekšējās ir operācijas (darbības). Binārajā kokā darbības attēlo ar šādiem simboliem: **„ + ”; „ - ”; „ \* ”; „ / ” ; „ ↑ ”.** Eksistē trīs veidu izteiksmju pierakstu formas: prefikss, infikss, postfikss.

Izteiksmes prefiksa pierakstu formu iegūst veicot kokam pirms sakārtojuma apiešanu. Prefiksa formu var novērtēt, ejot no labās uz kreiso pusi. Operācija (darbības zīme) atrodas pirms diviem operandiem. Lai iegūtu prefiksa formas vērtību, izpilda operāciju, ja operators atrodas pirms diviem operandiem.

Aplūkosim, piemēru, dota izteiksme ((2^(9-7)) + (9/(4-1)), izteiksmes attēlojums bināra koka formā ir šāds:

Izteiksmes prefiksa formas vērtību iegūsim vispirms veicot pirms sakārtojumu apiešanu, tad iegūtai izteiksmei, ejot no labās uz kreisu pusi, veiksim operāciju darbības ar operandiem līdz iegūsim gala rezultātu.



4.solis

5.solis

6.solis

1.solis

2.solis

3.solis

7.solis

8.solis

10.solis

9.solis

11.solis

1. apiešanas solī iegūstam operāciju „ + ”,
2. apiešanas solī iegūstam operāciju „^ ” ,
3. apiešanas solī iegūstam operandu 2,
4. apiešanas solī iegūstam operāciju „ - ”,
5. apiešanas solī iegūstām operandu 9,
6. apiešanas solī iegūstam operandu 7,
7. apiešanas solī iegūstām operāciju „ / ”,
8. apiešanas solī iegūstām operandu 9,
9. apiešanas solī iegūstam operāciju „-”,
10. apiešanas solī iegūstam operandu 4,
11. apiešanas solī iegūstam operandu 1.

Iegūstām izteiksmi + ^ 2 – 97 / 9 - 41, kas, savukārt, atbilst prefiksa pieraksta formai, turpmāk, ejot no labās uz kreisu pusi, līdz sasniegsim kādu no darbības zīmēm un veiksim atbilstošu darbību ar operandiem, aprēķināsim izteiksmes vērtību.

+ ^ 2 - 9 7 / 9 - 4 1

+ ^ 2 - 9 7 / 9 3

+ ^ 2 - 9 7 3

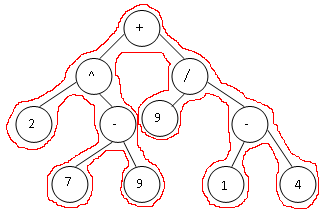
+ ^ 2 2 3

+ 4 3

**7**

Iegūtas izteiksmes rezultāts ir 7.

Izteiksmes postfiksa pieraksta formu iegūst veicot kokam pēcsakārtojuma apiešanu. Šajā formā operators seko aiz saviem diviem operandiem. Lai aprēķinātu postfiksa formas vērtību, iet no kreisās uz labo pusi, izpildot operāciju, ja operators seko diviem operandiem.

Iegūsim postfiksa pieraksta formu no iepriekšēja piemēra dotas izteiksmes ((2^(9-7)) + (9/(4-1)), veiksim pēcsakārtojuma apiešanu.

10.solis

8.solis

3.solis

11.solis

5.solis

4.solis

1.solis

2.solis

9.solis

6.solis

7.solis

1. apiešanas solī iegūstam operandu 7,
2. apiešanas solī iegūstam operandu 9,
3. apiešanas solī iegūstam operāciju „ - ”,
4. apiešanas solī iegūstam operandu 2,
5. apiešanas solī iegūstam operāciju „ ^ ”,
6. apiešanas solī iegūstam operandu 1,
7. apiešanas solī iegūstam operandu 4,
8. apiešanas solī iegūstam operāciju „ - ”,
9. apiešanas solī iegūstam operandu 9,
10. apiešanas solī iegūstam operāciju „ / ”,
11. apiešanas solī iegūstam operāciju „ + ”.

Iegūstām izteiksmi 79 – 2 ^ 14 – 9 / +, izteiksme atbilst postfiksa pieraksta formai, ejot no kreisas uz labo pusi, līdz sasniegsim kādu no darbības zīmēm un veiksim atbilstošu darbību ar operandiem, aprēķināsim izteiksmes vērtību.

79 - 2 ^ 1 4 - 9 / +

2 2 ^ 1 4 - 9 / +

4 1 4 - 9 / +

4 3 9 / +

43 +

**7**

Iegūtas izteiksmes rezultāts ir 7.

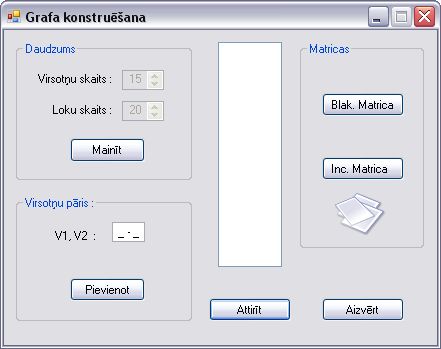
Izteiksmes infiksa pieraksta formu iegūst veicot kokam iekšējā sakārtojuma apiešanu.

# Paskaidrojumi programmas lietotājam

## 1.uzdevumam

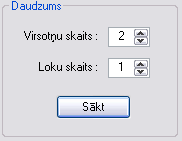
### Darba uzsākšana

Lietotājam ir jāpalaiž atkompilēto programmu „Matrix” ar paplašinājumu .exe. Pēc programmas palaišanas ekrānā tiek atvērta forma, kuras galvenais uzdevums ir veikt grafa konstruēšanu ( att. 3.1.1.1).



* + - 1. Att. Galvenā programmas forma.

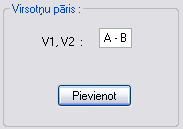
Lai uzsāktu darbu ar programmu, lietotājam vispirms ir jānorāda virsotņu un loku skaits grafā, citādi turpmākas darbības ar grafu nebūs pieejamas. Grafa virsotņu un loku skaits ir jānorāda speciāla tām atvelētāja lauka un jānospiež poga ar uzrakstu „Sākt” ( att. 3.1.1.2).



* + - 1. Att. Grafa virsotņu un loku uzdošana.

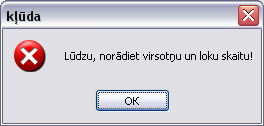
Uzreiz pēc minētas pogas nospiešanas uz pogas parādās uzraksts „Mainīt” un abi ievadlauki paliek neaktīvi (att. 3.1.1.1). Programma ļauj maksimāli uzdod 15 virsotnes un 20 lokus un minimums 2 virsotnes un 1 loku.

Turpmāk lietotājam ir jāievada grafa loka virsotņu pāris, tas ir izdarāms formas atvēlētāja loga ar nosaukumu „virsotņu pāris” (att. 3.1.1.3) .



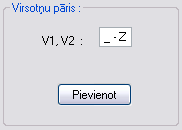
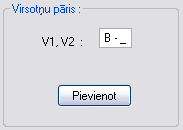
* + - 1. Att. Virsotnes pāra ievade.

Virsotņu pāris ir jāievada speciāla teksta laukā, kurš atļauj veikt vienīgi burtu ievadi, cipari vai citi simboli tiek ignorēti. Jā lietotājs nav norādījis virsotņu un loku skaitu, tad programma izvada brīdinājuma paziņojumu ar lūgumu norādīt virsotņu un loku skaitu ( att. 3.1.1.4).

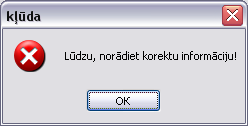


* + - 1. Att. Brīdinājuma paziņojums.

Jā lietotājs mēģina ievadīt nekorekto informāciju (att. 3.1.1.5 un 3.1.1.6) vai arī to neievada un nospiež pogu „Pievienot”, tad programma izvada brīdinājuma paziņojumu ar lūgumu ievadīt korektu informāciju (att. 3.1.1.7).



3.1.1.5 Att. un 3.1.1.6 Att. Nekorekta informācija.



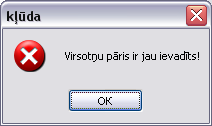
3.1.1.7 Att. brīdinājuma paziņojums.

Jā lietotājs mēģina ievadīt atkārtoti to pašu virsotni, tad programma izvada brīdinājuma paziņojumu, ka virsotne nevar būt refleksīva jeb uz galvenās diagonāles jābūt 0 (att. 3.1.1.8).



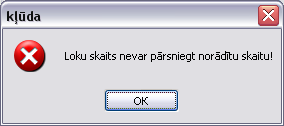
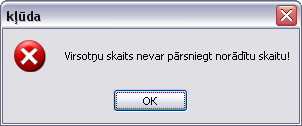
3.1.1.8 Att. Brīdinājuma paziņojums.

Jā lietotājs ievada atkārtotu virsotņu pāri, kas ir jau saraksta programma izvada kļūdas paziņojumu, ka esošais virsotņu pāris ir jau ievadīts (att. 3.1.1.9).



3.1.1.9 Att. kļūdas paziņojums.

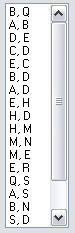
Lietotājam ir jāievada tik daudz loku un virsotņu skaitu, cik viņš to ir norādījis. Jā lietotājs ievada lielāku virsotņu vai loku skaitu, tad programma izvada brīdinājuma paziņojumus ar atbilstošam norādēm ( att. 3.1.1.10 un 3.1.1.11).



3.1.1.10 un 3.1.1.11 Att. kļūdas paziņojumi.

### Blakus virsotnes un incidences matricas formas

Ja lietotājs ir ievadījis korekto informāciju un nospiedis pogu „Pievienot”, tad virsotņu pārim ir jāparādās formas sarakstā (att. 3.1.2.1). Sarakstām apakša ir poga „Attīrīt”, ja poga tiek nospiesta tad saraksts tiek attīrīts ( att. 3.1.1.1).



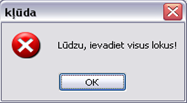
3.1.2.1 Att. Saraksts ar pievienotām virsotnēm.

Pēc virsotņu ievades, lietotājs var izvelēties, kuru no matricām programma izvadīs vispirms. Tas ir izdarāms loga „Matricas” ( att. 3.1.2.2).



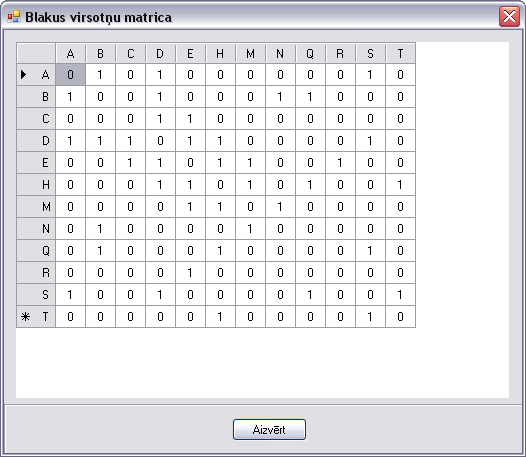
3.1.2.2 Att. Izvēles logs.

Jā lietotājs mēģina izvadīt kādu no matricu formām, bet loku skaits ir nepietiekams, tad programma izvada kļūdas paziņojumu ar norādi, ka lietotājam vispirms ir jānorāda pilnais loku skaits (att. 3.1.2.3).



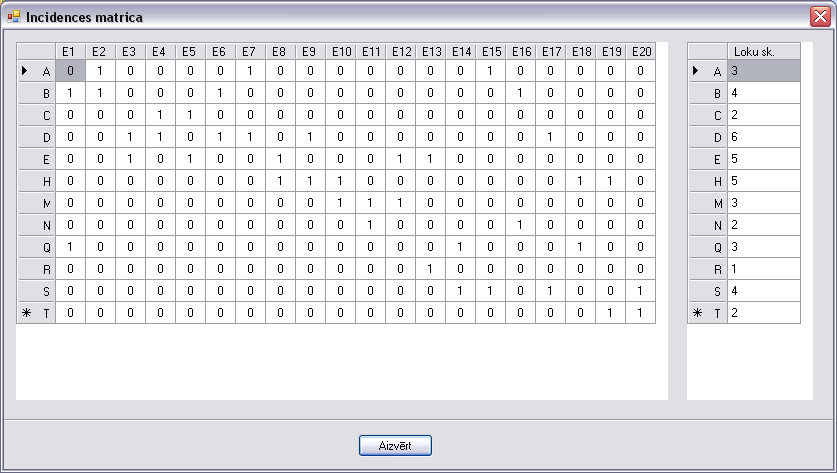
3.1.2.3 Att. kļūdas paziņojums.

Pēc pogas „Blak. Matrica” nospiešanas atveras blakus virsotņu matricas forma, kurā ir redzama blakus virsotņu matrica (att. 3.1.2.4).



3.1.2.4 Att. Blakus virsotņu matricas forma.

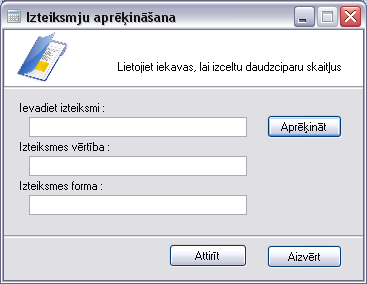
Pēc pogas „Inc. Matrica” nospiešanas atveras incidences matricas forma, kurā ir redzama incidences matrica un lokālo virsotņu skaits katrai virsotnei (att. 3.1.2.5).



Att. 3.1.2.5 Incidences matricas forma.

## 2.uzdevumam

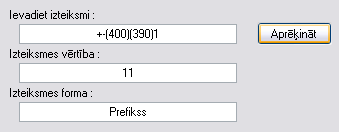
Lietotājam ir jāpalaiž atkompilēto programmu „Expressions” ar paplašinājumu .exe. Pēc programmas palaišanas ekrānā tiek atvērta galvenā forma, kuras galvenais uzdevums ir veikt ievadīto izteiksmju aprēķināšanu un rezultāta izvadi ( att. 3.2.1).



3.2.1 Att. Programmas galvenā forma.

Programma satur trīs ievades laukus, pirmajā ievades laukā lietotājam ir jāievada izteiksmi, pārējie ievades lauki izvada rezultātu. Programma satur pogu „Attīrīt”, pēc pogas nospiešanas visi ievadlauki tiek attīrīti jeb informācija, kas ir izvadīta vai ievadīta ievadlaukos tiek izdzēsta.

Lai iegūtu izteiksmes rezultātu lietotājam ir jāveic izteiksmes ievade pirmajā ievades laukā un jānospiež poga „Aprēķināt”, tad otrajā ievades lauka parādās izteiksmes rezultāts un trešajā izteiksmes forma, t.i., prefikss vai postfikss (att. 3.2.2).



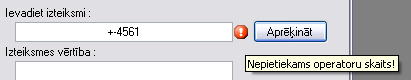
3.2.2 Att. Rezultāta izvade.

Jā lietotājs nospiež pogu „Aprēķināt”, bet pirms tām nav ievadījis izteiksmi, tad programma izvada izsaukuma zīmi un tekstu ar lūgumu vispirms ievadīt izteiksmi (att. 3.2.3).



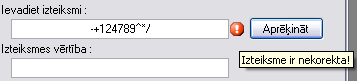
3.2.3 Att. Brīdinājuma izvade.

Jā lietotājs ievada izteiksmi, kura operandu skaits ir daudzkārt lielāks, par operāciju skaitu, vai arī otrādi, tad programma izvada atbilstošu paziņojumu (att. 3.2.4).



3.2.4 Att. Brīdinājuma izvade.

Jā lietotājs mēģina ievadīt nekorekto izteiksmi, tad programma izvada paziņojumu, ka ievadīta informācija ir nekorekta (att. 3.2.5).



3.2.5 Att. Brīdinājuma izvade.

# Kontroles piemēra analīze

## 1.uzdevumam

Dots šāds neorientēts grafs (skat. zemāk) :

Uzrakstīsim grafa loku virsotņu pārus: E1 (a, b), E2 (b, c), E3 (c, d), E4 (d, e), E5 (e, h), E6 (h, m), E7 (m, n), E8 (n, q), E9 (q, r), E10 (r, a), E11 (q, s), E12 (s, t), E13 (t, e), E14 (b, s), E15 (r, s), E16 (t, n), E17 (t, m).

Grafam ir 12 virsotnes un 17 loki, sastādot grafa blakus virsotņu matricu iegūstam sekojošo kvadrātveida matricu:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | H | M | N | Q | R | S | T |
| A | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 |
| B | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
| C | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** |
| H | 0 | 0 | 0 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 |
| M | 0 | 0 | 0 | 0 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | **1** |
| N | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | **1** | 0 | 0 | **1** |
| Q | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | **1** | **1** | 0 |
| R | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | **1** | 0 |
| S | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | **1** |
| T | 0 | 0 | 0 | 0 | **1** | 0 | **1** | **1** | 0 | 0 | **1** | 0 |

Sastādot grafa incidences matricu iegūstam sekojošo taisnstūrveida matricu:

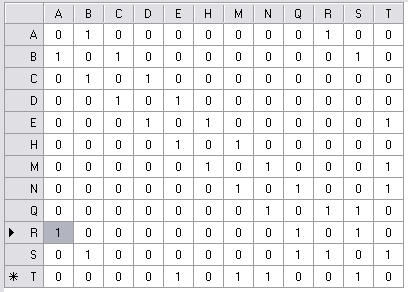
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | E12 | E13 | E14 | E15 | E16 | E17 |
| A | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| C | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 |
| H | 0 | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| M | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| N | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
| Q | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | **1** | 0 | 0 |
| S | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | **1** | **1** | 0 | 0 |
| T | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 | 0 | **1** | **1** |

Virsotņu lokālas pakāpes ir šādas:

|  |  |
| --- | --- |
|  | Virsotņu lokālas pakāpes |
| A | 2 |
| B | 3 |
| C | 2 |
| D | 2 |
| E | 3 |
| H | 2 |
| M | 3 |
| N | 3 |
| Q | 3 |
| R | 3 |
| S | 4 |
| T | 4 |

Programmas „Matrix” rezultāts:

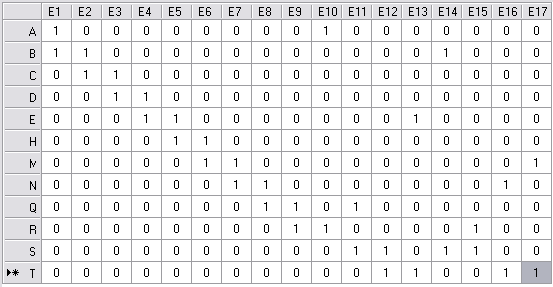
1. Blakus virsotņu matrica (att. 4.1.1):



* + 1. Att. Blakus virsotņu matrica.

Salīdzinot, ar manis iepriekš konstruētu blakus virsotņu matricu ( 19.lpp ), matricas ir pilnīgi identiskās, secinājums programma strādā korekti.

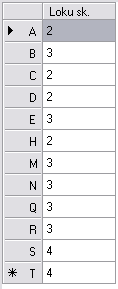
1. Incidences matrica (att. 4.1.2):



4.1.2 Att. Incidences matrica.

Salīdzinot, ar manis iepriekš konstruētu incidences matricu ( 20. lpp ), matricas ir pilnīgi identiskās, secinājums programma strādā korekti.

1. Virsotņu lokālas pakāpes (att. 4.1.3) :



* + 1. Att. Virsotņu lokālas pakāpes.

Salīdzinot, ar manis iepriekš iegūtam matricas virsotņu lokālajām pakāpēm ( 20. lpp ), rezultāts ir pilnīgi identisks, secinājums programma strādā korekti.

## 2.uzdevumam

Aprēķināsim dotas prefiksa pieraksta izteiksmes vērtību:

+ / / ^ + 134 / - (50)2343

+ / / ^ + 134 / (48)343

+ / / ^ + 134(16)43

+ / / ^ 44(16)43

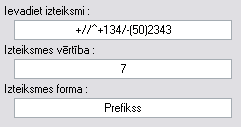
+ / / (256)(16)43

+ / (16)43

+ 43

7

Programmas „Expressions” dotas izteiksmes iegūtais rezultāts (att. 4.2.1):



4.2.1 att. programmas izpildes rezultāts.

Salīdzinot, ar manu roku iegūtu un programmas aprēķinātu rezultātu izteiksmes vērtības sakrīt, secinājums programma strādā korekti.

Aprēķināsim dotas postfiksa pieraksta izteiksmes vērtību:

372 + ^ 5(10) - + 34(10) \* + +

39 ^ 5(10) - + 34(10) \* + +

(729)5(10) - + 34(10) \* + +

(729)5 + 34(10) \* + +

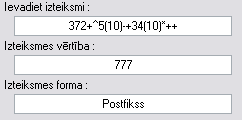
(734)34(10) \* + +

(734)3(40) + +

(734)(43) +

777

Programmas „Expressions” dotas izteiksmes iegūtais rezultāts (att. 4.2.2):



4.2.2. att. programmas izpildes rezultāts.

Salīdzinot, ar manu roku iegūtu un programmas aprēķinātu rezultātu izteiksmes vērtības sakrīt, secinājums programma strādā korekti.

# Secinājumi

Studiju darba izstrādes gaita tika detalizēti izanalizēts un izprasts teorētiskais materiāls par incidences un blakus virsotņu matricām, izteiksmju pieraksta formām. Iegūtas zināšanas ļāva izstrādāt divas programmas atbilstoši uzdevumu nosacījumu prasībām.

Runājot par uzdevuma sarežģītības pakāpēm, pirmais uzdevums prasīja daudz ilgāk laika gan izstrādājot programmu, gan rakstot teorētisko pamatojumu.

Saistība ar studiju darba izstrādes procesu, tas bija interesants un saistošs, tas rosināja un attīstīja loģisko domāšanu, bet gribētos vēl padziļinātak izprast grafu realizāciju datorzinātnes, saistība gan ar datu bāzēm, gan arī ar datorgrafiku.

Uzskatu, ka esmu papildinājis savas zināšanas mācību priekšmeta „Diskrētas struktūras datorzinātnēs”.

# Literatūras saraksts

1. Lekciju konspekts mācību priekšmetā „Diskrētas struktūras datorzinātnes”, prof. Grundspeņķis, RTU, 2001/2002 m.g.
2. Дарья Шевякова, Андрей Степанов, Андрей Карпов «Самоучитель Visual Basic 2005», Санкт-Петербург, «БХВ-Петербург», 2007.
3. Internets: [http://en.wikipedia.org/wiki/Graph\_theory#Matrix\_structures](http://en.wikipedia.org/wiki/Graph_theory" \l "Matrix_structures)

Internets: [http://ru.wikipedia.org/wiki/](http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)#.D0.9C.D0.B0.D1.82.D1.80.D0.B8.D1.86.D0.B0_.D1.81.D0.BC.D0.B5.D0.B6.D0.BD.D0.BE.D1.81.D1.82.D0.B8)

Internets: <http://en.wikipedia.org/wiki/Tree_data_structure#Root_nodes>

Internets: <http://www.cs.man.ac.uk/~pjj/cs2121/fix.html>