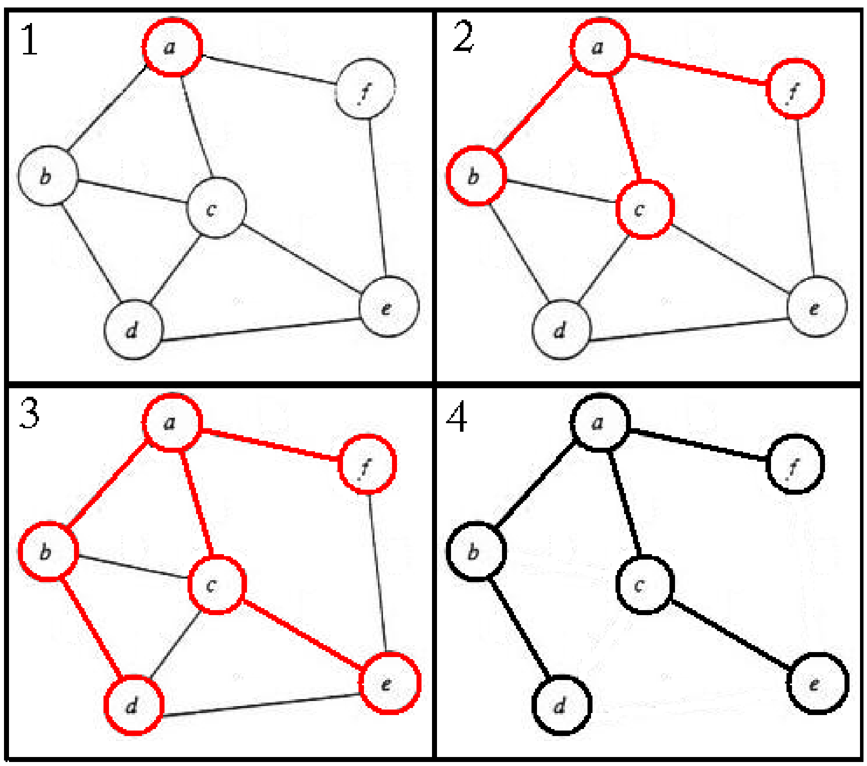
KETURIOLIKTAS SKYRIUS

Algoritmai su medžiais

Dalis svarbiausių grafų teorijos sąvokų, algoritmų, uždavinių klasių, kurių sprendimui taikoma grafų teorija, aprašyti P. Tannenbaumo ir R. Arnoldo knygoje „Kelionės į šiuolaikinę matematiką“ (Vilnius: TEV, 1995). Medžiai nagrinėjami šios knygos 7 skyriuje.

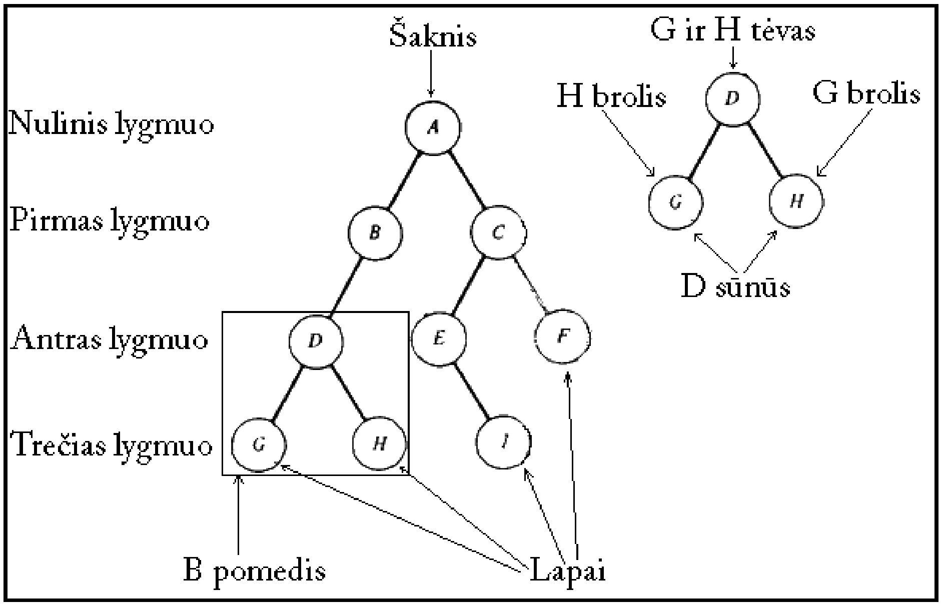
Pagrindinės sąvokos

Medis grafų teorijoje – tai jungus neorientuotas grafas, neturintis ciklų. Toks apibrėžimas parodo, kad medis, kaip ir grafas, yra objektų ir jų tarpusavio sąryšių visuma. Akivaizdu, kad bet kokias dvi skirtingas medžio viršūnes jungia vienintelis kelias. Todėl visiems medžiams egzistuoja griežta, vienareikšmė atitiktis tarp jo viršūnių ir briaunų skaičių. Taip pat pastebėtina, kad, jungiame neorientuotame grafe atlikus paiešką platyn arba paiešką gilyn, pografis (pradinio grafo jungčių aibė), sudarytas iš nagrinėtų kelių, yra medis. Egzistuoja net atskira uždavinių, kuriuose ieškomas tam tikrų savybių turintis grafo medis, klasė ir algoritmai jiems spręsti.



Daugiau uždavinių yra būtent su medžiais, nei medžių išskyrimo iš grafų: naudingiau medį nagrinėti kaip hierarchinę duomenų struktūrą, nei atskirą grafo atvejį ar pografį. Hierarchijai apibrėžti viena medžio viršūnė išskiriama ir vadinama šaknimi. Kelias iš šaknies į kurią nors viršūnę vadinamas šaka. Medis braižomas šaknimis žemyn, šaknis pateikiama viršuje.

Nors briaunos paprastai neturi krypties, intuityviai joms priskiriama kryptis. Jei ši kryptis rodo į kitą viršūnę, esančią žemyn nuo šaknies, tai sakoma, kad viršūnė turi sūnų. Briaunos kryptis į priešingą pusę (viršų) nurodo viršūnei jos tėvą. Viršūnė be sūnų vadinama lapu arba išorine viršūne. Visos kitos viršūnės vadinamos vidinėmis. Vieno tėvo sūnūs vadinami broliais. Jei kiekvienas tėvas turi ne daugiau kaip fiksuotą skaičių sūnų (n), jis vadinamas n-ariniu medžiu. Medis, prasidedantis kurioje nors viršūnėje, vadinamas pomedžiu. Medis turi lygius: šaknis yra nulinio lygio, jos sūnūs – pirmo, šių sūnų sūnūs – antro ir t. t. Medžio aukščiu vadinama didžiausio lygio reikšmė. Miškas yra nemažiau kaip dviejų medžių aibė.



Programavime ypač svarbūs yra dvejetainiai medžiai. Dažniausiai išskiriamas pilnas, tobulas ir subalansuotas dvejetainis medis. Pilnas dvejetainis medis – medis, kurio kiekviena viršūnė arba neturi sūnų, arba turi du sūnus. Tobulas dvejetainis medis – pilnas dvejetainis medis, kurio visi lapai yra viename lygmenyje. Subalansuotas dvejetainis medis – dvejetainis medis, kuriame visi lapai yra dvejuose gretimuose lygmenyse.

Medžio vaizdavimas kompiuteryje

Medžio vaizdavimo kompiuteryje būdai iš esmės skiriasi nuo grafo vaizdavimo, nors medį, be abejo, galima laikyti grafu bei vaizduoti kaip grafą. Bet stengiamasi pabrėžti medžio hierarchinę struktūrą ir išnaudoti jos teikiamus privalumus sprendžiant iškeltą uždavinį. Tokiu būdu uždavinys gali įtakoti vaizdavimo būdo pasirinkimą. Dažniausiai išskiriami keturi medžio vaizdavimo būdai:

a) naudojant masyvus;
b) naudojant dinamines duomenų struktūras (nuorodas);
c) naudojant metodą „kairysis sūnus – dešinysis brolis“;
d) naudojant rekursyvųjį masyvą.

A. Pavaizduokime dvejetainį medį, kurio viršūnių reikšmių tipas yra *T*. Aprašykime tris masyvus:

```
reikšmė : array [1..n] of T;
kairysis, dešinysis : array [1..n] of 0..n;
```

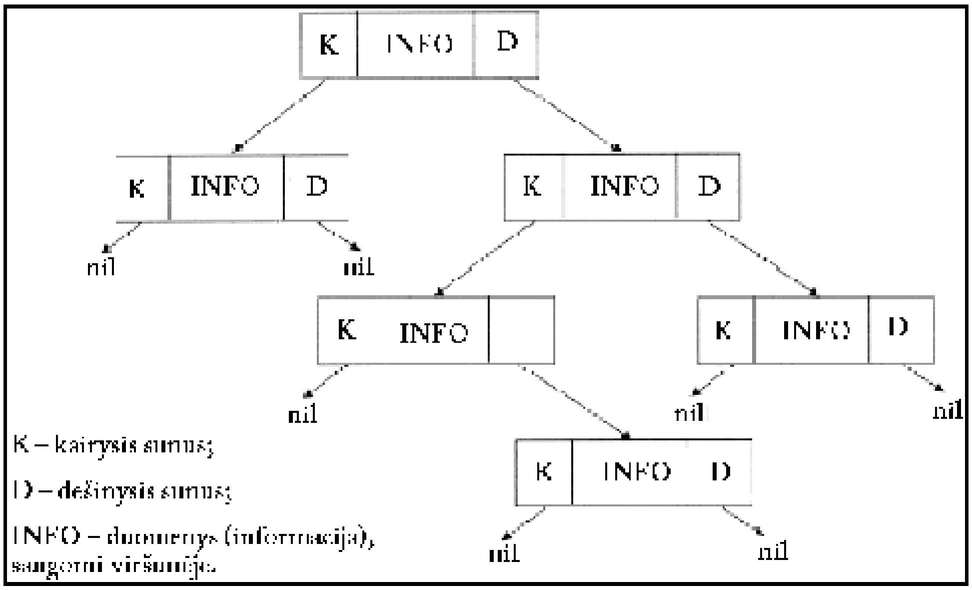
ir kintamąjį `šaknis : 0..n` (*n* – didžiausias galimas medžio viršūnių skaičius). Jei kintamojo `šaknis` reikšmė yra 0, medis tuščias. Kitu atveju viršūnės turi skirtingus numerius iš intervalo[1...*n*]. Viršūnės *i* reikšmė yra `reikšmė[i]`. Jei viršūnė *i* turi sūnų, tai jų numeriai yra `kairysis[i]` ir `dešinysis[i]`; `kairysis[i]` arba (ir) `dešinysis[i]` reikšmė 0 rodo, kad viršūnė *i* sūnaus(-ų) neturi.

Vietoj reikšmės 0, žyminčios neegzistuojančias viršūnes, galima naudoti bet kokią nepriklausančią intervalui [1...*n*] reikšmę. Dažnai vietoj 0 naudojama konstanta `null` = 0.

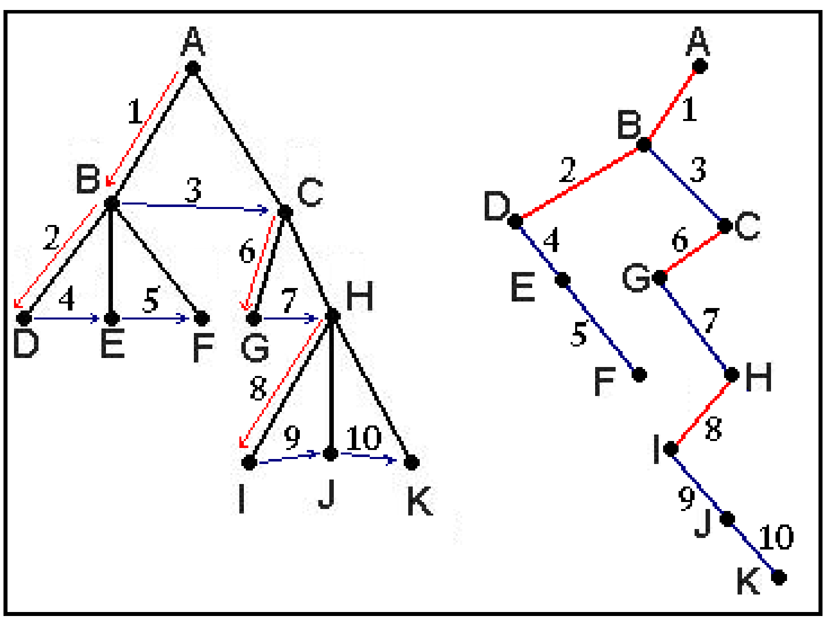
Esant *m*-nariui medžiui, sūnums žymėti naudojami ne du masyvai, o *m*. Arba dvimatis $m \times n$ matmenų masyvas. Esant mažam medžio šakotumui tai neefektyvu, nes daug elementų turi reikšmę `null`. Tada geriau parinkti kitą medžio vaizdavimo būdą arba net taikyti grafo vaizdavimo būdus.

B. Medį galima vaizduoti dinaminėmis duomenų struktūromis. Apie jas išsamiau galite paskaityti V. Tumasonio knygoje „Paskalis ir Turbo Paskalis 7.0“ (Vilnius:Ūkas, 1993; p. 134–150) ar kuriame kitame Paskalio kalbos žinyne. Tai vienas iš labiausiai paplitusių medžio vaizdavimo būdų. Viršūnės elementai saugomi įrašuose su nuorodomis. Nuorodos yra nukreiptos iš tėvo viršūnės į sūnus, nuorodų skaičius turi būti lygus didžiausiam galimam sūnų skaičiui. Šitokiu būdu dvejetainio medžio viršūnė aprašoma taip:

```
type T = ...;  {duomenų (informacijos), saugomų viršūnėje, tipas}
     Medis = ^viršūnė;
     viršūnė = record
                 INFO : T;
                 K, D : Medis;
               end;
```



C. Kaip ir B atveju, bet kuriam *n*-nariam medžiui galima naudoti įrašus tik su dviem nuorodomis: kairioji nuoroda identifikuoja viršūnės kairįjį sūnų, o dešinioji – viršūnės dešinį brolį. Toks vaizdavimo būdas reikalauja įvesti tam tikrą tvarką tarp brolių, kad atskirtume, kuris brolis yra kairėje, o kuris – dešinėje. Tokiu būdu didelio šakotumo medžius galima vaizduoti kaip dvejetainius:



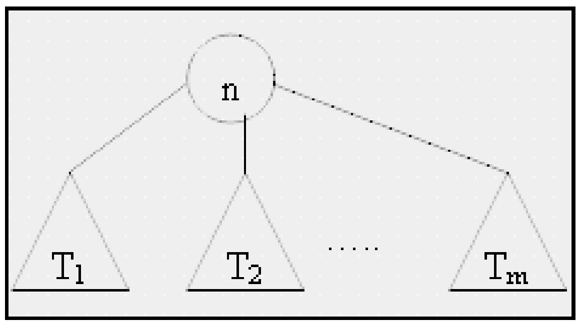
D. Rekursyvaus medžio metodas gali būti naudojamas tada, kai algoritme dažnai reikia naudoti nuorodas iš sūnų į jų tėvus. Medžio viršūnes sužymėjus pagal lygius, galima jas rašyti į vienmatį masyvą taip, kad masyvo indeksai atitiktų viršūnių žymėjimus, o elementų reikšmės būtų nuorodos į tėvo žymėjimus. Naudojant šį metodą galima labai lengvai „vaikščioti“ medžio šakomis nuo sūnų prie tėvų ir nuo vienos viršūnės prie kitos. Rekursyvaus medžio (pavaizduoto paveikslėlyje) vaizdavimo pavyzdys:

A	B	C	D	E	F	G	H	I	J	K
	A	A	B	B	B	C	C	H	H	H

Medžio numeracija

Uždaviniuose dažnai svarbūs sutvarkyti medžiai. Medis sutvarkytas, kai jo viršūnėms priskirta tam tikra numeracija (viršūnių apėjimo tvarka). Medžio viršūnės gali būti numeruojamos skirtingais būdais. Jei medis T turi šaknį n ir pomedžius T_1, T_2, ..., T_m, galimos tokios numeracijos:

- Prefiksinė tvarka. Pirmiausia aplankoma šaknis n, po to prefiksine tvarka aplankomos pomedžio T_1 viršūnės, vėliau – visos T_2 viršūnės ir t. t. Paskutinėmis prefiksine tvarka aplankomos pomedžio T_m viršūnės;
- Postfiksinė tvarka. Pirmiausia postfiksine tvarka apankamos visos pomedžio T_1 viršūnės, po to – pomedžių T_2, ..., T_m viršūnės. Paskutine aplankoma šaknis n;
- Infiksinė tvarka. Pirmiausia infiksine tvarka aplankomos visos pomedžio T_1 viršūnės, po to – šaknis n. Paskutinėmis infiksine tvarka apankamos pomedžių T_2, ..., T_m viršūnės.



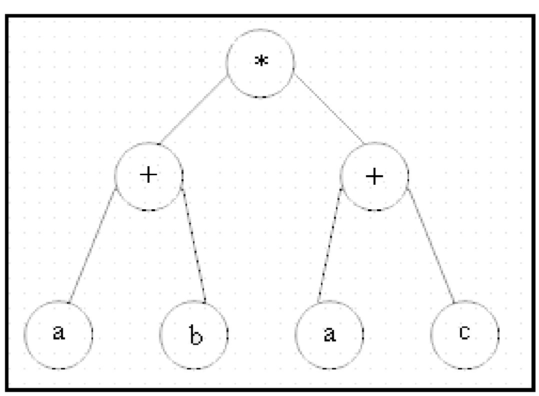
Tokios numeracijos dažnai naudojamos sintaksiniuose medžiuose. Sintaksiniai medžiai taikomi algebrinių, loginių reiškinių ir kitų sintaksinių struktūrų nagrinėjimui analizatoriuose, transliatoriuose. Jų dėka galima gauti prefiksines, postfiksines ir infiksines tokių reiškinių išraiškas.

Infiksinė reiškinio išraiška – įprastas reiškinio užrašymas su skliausteliais: $(a+b)*(a+c)$.

Prefiksinė išraiška konstruojama (rekursyviai) taip: operacijos ženklas, pirmas operandas, antras operandas: $*+ab+ac$.

Postfiksinė išraiška konstruojama (rekursyviai) taip: pirmas operandas, antras operandas, operacijos ženklas: $ab+ac+*$.

Infiksinėse išraiškose skliausteliai naudojami operacijų atlikimo tvarkai nurodyti. Prefiksinėse it postfiksinėse išraiškose skliausteliai nebūtini. Nagrinėjamą algebrinę išraišką atitinkantis sintaksinis medis:



Pastebėtina, kad operacijų ženklai yra vidinės medžio viršūnės, o operandai – išorinės. Todėl gali prireikti naudoti skirtingus tipus šių viršūnių reikšmėms aprašyti.

Dar vienas galimas medžio numeravimo būdas – viršūnių numeracija pagal lygius: šaknis; visos pirmo lygio viršūnės (numeruojamos iš kairės į dešinę); visos antro lygio viršūnės (numeruojamos ta pačia tvarka) ir t. t.

Uždaviniai

1. Dvejetainio medžio viršūnę aprašo toks įrašas:

```
type T = ...;
     Medis = ^viršūnė;
     viršūnė = record
               INFO : T;
               K, D : Medis;
            end;
```

Sukurta procedūra medžiui trinti:

```
procedure Trink(P: Medis);

  function ArLapas(P: Medis): boolean;
  begin
    ArLapas := (P^.K = nil) and (P^.D = nil);
  end;

  procedure TrinkLapą(var P: Medis);
  begin
    dispose(P); P := nil;
  end;

begin
  if (P^.K <> nil)
    then
      if ArLapas(P^.K) then TrinkLapą(P^.K) else Trink(P^.K);
  if (P^.D <> nil)
    then
      if ArLapas(P^.D) then TrinkLapą(P^.D) else Trink(P^.D);
  if ArLapas(P) then TrinkLapą(P);
end;
```

Tačiau procedūra veikia neteisingai: ištrinami tik lapai ir jų tėvai. Raskite klaidą ir paaiškinkite ją. Ištaisykite procedūrą, kad ji veiktų korektiškai – trintų visą medį.

2. *Parašykite loginę funkciją*, kurios rezultatas būtų `TRUE`, jei beciklis neorientuotas besvoris grafas yra medis, ir `FALSE`, jei miškas.

Grafo aprašymo būdą JPM klausytojas parenka pats.

3. Parašykite reiškinių postfiksines ir prefiksines išraiškas bei nupieškite jas atitinkančius sintaksinius medžius:

a) $((A-C)+F*(D+B))*E$;

b) $(A+(B+(C+C*F)))-E*A$;

c) $(B+A)-(E+C)+A*(E+A)$;

d) $A*(E+E)*A+(B-A*E)-A$;

e) $(A*(B-D))*E-C*(D+(E*F+G))$.

4. *Parašykite funkciją*, nustatančią maksimalų medžio gylį. Prieš funkciją komentaruose turi būti nurodytas medžio vaizdavimo būdas atmintinėje.

Šio skyriaus 1–4 uždavinių sprendimai turi būti pateikti JPM interneto svetainėje (http://ims.mii.lt/jpm/) iki 2009 m. gegužės 03 d. 24 val.

Failo vardas turi būti *?????14.** (čia * – *doc, txt, odt arba rtf*). Vietoj klaustukų įrašykite pirmąsias penkias savo pavardės raides (be diakritinių ženklų). Jei kartais būtų pavardžių, trumpesnių negu penkios raidės, trūkstami simboliai keičiami pabraukimo brūkšniais „_“.

Elektroninio pašto adresas JPM antrosios dalies klausytojams: jpm.2kursas@gmail.com.