$\alpha + \omega$, 2001, Nr. 3, 88–97

Maple – naujos galimybės

Aleksas Domarkas
aleksas@ieva.mif.vu.lt

Straipsnio autorius, VU Matematikos ir informatikos fakulteto docentas, demonstruoja kompiuterinės algebros sistemos Maple galimybes sprendžiant nelengvus matematikos ir informatikos uždavinius.

Kompiuterinės algebros sistema Maple atveria naujas galimybes studijuojantiems ir taikantiems matematiką. Iš dalies su šiomis galimybėmis jūs galite susipažinti peržiūrėję čia pateiktus uždavinių sprendimo pavyzdžius, kurie atlikti Maple 6 programos versija. Daugiau pavyzdžių galite rasti [4] darbe. Sparčiai tobulėjant kompiuterinei technikai ir programinei įrangai, šios galimybės nuolat didėja.

Kiekvieno studijuojančio matematiką ar atliekančio matematinius skaičiavimus darbo vietoje turėtų būti kompiuteris su keliomis matematinio skaičiavimo sistemomis. Svarbiausios iš jų yra Maple, Mathematica, Macsyma, Matlab, Mathcad, Mupad, Derive ir kt. Platesnę jų apžvalgą ir palyginimą galima rasti didelės apimties [5] knygoje. Kokią sistemą pasirinkti, spręskite patys. Susipažinus su viena sistema, nesunku pereiti prie kitos. Kompiuterinės algebros sistemos gali vykdyti ne tik tradicinius skaičiuoklio veiksmus, bet kur kas daugiau: pertvarkyti aritmetinius ir algebrinius reiškinius, rasti baigtines ir begalines sumas bei sandaugas, apskaičiuoti išvestines ir integralus, spręsti lygtis ir lygčių sistemas, atlikti veiksmus su vektoriais ir matricomis, braižyti įvairius grafikus ir pan. Šias operacijas vykdyti nėra sudėtinga, nes reikia tik susirasti tinkamą komandą. Maple 6 sistemoje tokių komandų yra apie 3000. Sudėtingesniems ir nestandartiniams uždaviniams spręsti tenka pasinaudoti programavimo galimybėmis, jungiant komandas į programą — komandų seką arba apibrėžiant naujas komandas-procedūras (žr. toliau pateikiamus uždavinių sprendimo pavyzdžius).

Keletas nurodymų tiems, kurie nagrinės pavyzdžius. Kiekviena Maple komanda baigiasi kabliataškiu arba dvitaškiu. Paspaudus Enter, komanda abiem atvejais yra vykdoma, tik dvitaškio atveju rezultatas nerodomas ekrane. Ženklas := yra vardo suteikimo operacija. Kairėje šio ženklo mes rašome vardą, kurį parenkame objektui. Ženklas % yra paskutinės komandos vykdymo rezultatas. Apie kiekvieną komandą galima sužinoti iškvietus pagalbą. Paprasčiausias iškvietimo būdas — pelės žymeklį nustatyti ties komandos pavadinimu ir paspausti klavišą F1. Pagalbą taip pat galima iškviesti surinkus ženklą ? ir komandos pavadinimą, pavyzdžiui,

```
> ?solve
```

Manau, kad šis darbas jus sudomins ir padės išsirinkti tinkamą kompiuterinės algebros sistemą.

1 uždavinys (žr. [1]). *Kiekvienam $n \geqslant 1$ raskite n-ojo laipsnio daugianarį $P_n(x)$, su visais $t \neq 0$ tenkinantį sąlygą*

$$P_n\left(t + \frac{1}{t}\right) = t^n + \frac{1}{t^n}.$$

Pirmas būdas. Naudojamės neapibrėžtųjų koeficientų metodu. Daugianarių ieškome pavidalo

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n.$$

Koeficientus galima rasti komanda `solve(identity(eqn,x), vars)` (žr. `solve/identity`).

```
> restart;
> ?solve/identity
```

Sudarome procedūrą T1(n) n-ojo laipsnio daugianariui rasti:

```
> T1:=proc(n) local k,f,Ta; f:=x->sum(a[k]*x^k,k=0..n);
Ta:=f(t+1/t)=t^n+1/t^n; solve(identity(Ta,t),{seq(a[k],k=0..n)});
expand(subs(%,f(x))); RETURN(sort(%)); end proc:
```

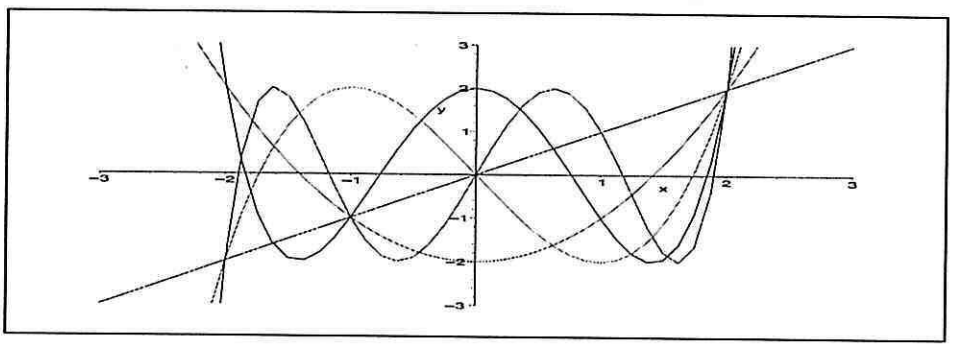
Pirmieji penki daugianariai yra:

```
> seq(T1(k),k=1..5);
```

$$x, \quad x^2-2, \quad x^3-3x, \quad x^4-4x^2+2, \quad x^5-5x^3+5x$$

Pavaizduosime juos grafiškai (1 pav.):

```
> plot([%],x=-3..3,y=-3..3);
```



1 pav.

Nesunku pastebėti, kad $P_n(2) = 2$ ir su kiekvienu x iš $[-2; 2]$ teisinga nelygybė $|P_n(x)| \leqslant 2$.

Antras būdas. Darbe [1] yra išvestas rekurentusis sąryšis

$$P_{n+1}(x) = P_n(x)x - P_{n-1}(x).$$

Iš šio sąryšio komanda `rsolve` galima rasti n-ojo daugianario formulę:

```
> rsolve( {f(n+1)=f(n)*x-f(n-1),f(1)=x,f(2)=x^2-2},f);
```

$$\left(2\cdot\frac{1}{x+\sqrt{x^2-4}}\right)^n + \left(-2\cdot\frac{1}{-x+\sqrt{x^2-4}}\right)^n$$

Kiekvieno dėmens vardiklyje panaikiname iracionalumą:

$\alpha + \omega$

```
> map(rationalize,op(1,%))+map(rationalize,op(2,%));
```

$$\left(\tfrac{1}{2}x-\tfrac{1}{2}\sqrt{x^2-4}\right)^n+\left(\tfrac{1}{2}x+\tfrac{1}{2}\sqrt{x^2-4}\right)^n$$

```
> g:=unapply(%,n);
```

$$g:=n\to\left(\tfrac{1}{2}x-\tfrac{1}{2}\sqrt{x^2-4}\right)^n+\left(\tfrac{1}{2}x+\tfrac{1}{2}\sqrt{x^2-4}\right)^n$$

```
> T2:=n->simplify(g(n));
```

$$T2:=n\to \text{simplify}(g(n))$$

Gavome formulę

$$P_n(x)=\left(\frac{x}{2}-\frac{\sqrt{x^2-4}}{2}\right)^n+\left(\frac{x}{2}+\frac{\sqrt{x^2-4}}{2}\right)^n.$$

Toliau pagal tą formulę apskaičiuojame pirmuosius penkis daugianarius:

```
> seq(T2(k),k=1..5);
```

$$x,\quad x^2-2,\quad x^3-3x,\quad x^4-4x^2+2,\quad x^5-5x^3+5x$$

Trečias būdas. Apibrėžiame procedūrą, pagal kurią rekurentiškai apskaičiuojami daugianariai.

```
> T3:=proc(n) option remember; expand(T3(n-1)*x-T3(n-2));
end proc:
> T3(1):=x:  T3(2):=x^2-2:
> seq(T3(k),k=1..5);
```

$$x,\quad x^2-2,\quad x^3-3x,\quad x^4-4x^2+2,\quad x^5-5x^3+5x$$

Palyginame skaičiavimų šiais būdais (naudojantis 600 Mz Pentium III) greitį. Randame laiką, per kurį apskaičiuojamas 50-asis daugianaris:

```
> time(T1(50));
                              5.078
> time(T2(50));
                               .173
> time(T3(50));
                               .004
```

Matome, kad pats greičiausias yra trečiasis būdas. Lėčiausias yra pirmasis būdas, bet jis nenaudoja rekurenčiojo sąryšio.

2 uždavinys. *Tegul a, b, c — teigiamieji skaičiai, $abc=1$. Įrodykite, kad*

$$\left(a-1+\frac{1}{b}\right)\left(b-1+\frac{1}{c}\right)\left(c-1+\frac{1}{a}\right)\leqslant 1.$$

Šis uždavinys pateiktas 41-ojoje pasaulinėje olimpiadoje ir buvo sunkiai įkandamas (žr. [2]). Keli jo sprendimo būdai išnagrinėti [2] darbe. Pateiksime dar vieną sprendimo variantą. Naudojantis Maple, šis būdas yra gana paprastas.

Atliekame keitinį $a=\mathrm{e}^x$, $b=\mathrm{e}^y$, $c=\mathrm{e}^z$. Gauname ekvivalentų uždavinį:

Tegul x, y, z — bet kurie realieji skaičiai, $x+y+z=0$. Įrodykite, kad

$$\left(\mathrm{e}^x-1+\mathrm{e}^{(-y)}\right)\left(\mathrm{e}^y-1+\mathrm{e}^{(-z)}\right)\left(\mathrm{e}^z-1+\mathrm{e}^{(-x)}\right)\leqslant 1.$$

Šitaip atsikratome papildomos sąlygos, kad kintamieji yra teigiamieji skaičiai. Toliau taikome paprasčiausią metodą — išreiškiame kintamąjį $z = -x - y$ ir įstatome į kairiąją nelygybės pusę. Tada nelygybei įrodyti pakanka išspręsti standartinį uždavinį — rasti dviejų kintamųjų funkcijos

$$f = \left(e^x - 1 + e^{(-y)}\right)\left(e^y - 1 + e^{(x+y)}\right)\left(e^{(-x-y)} - 1 + e^{(-x)}\right)$$

maksimumą. Be kompiuterio tai įveikti sunku dėl funkcijos f ir jos išvestinių sudėtingumo. Maksimumui rasti naudinga žinoti *ekstremumo nustatymo taisyklę*:

1. Kritiniams taškams $M(x; y)$ nustatyti reikia išspręsti lygčių sistemą $f_x = 0$, $f_y = 0$.
2. Kiekviename kritiniame taške M reikia apskaičiuoti antrąsias išvestines f_{xx}, f_{yy}, f_{xy} ir reiškinį $\Delta = f_{xx}f_{yy} - f_{xy}^2$.
3. Jei $\Delta > 0$, tai M yra ekstremumo taškas: kai $f_{xx} < 0$ — maksimumas, kai $f_{xx} > 0$ — minimumas. Jei $\Delta < 0$, tai ekstremumo taške M nėra, o jei $\Delta = 0$, tai reikia tirti papildomai.

```
> restart;
```

Apibrėžiame kintamųjų keitimo formules:

```
> T:={a=exp(x),b=exp(y),c=exp(z)}:
```

Įstatome į sąryšio sąlygą:

```
> subs(T,a*b*c=1);
```

$$e^x e^y e^z = 1$$

Šią lygybę logaritmuojame:

```
> map(ln,%);
```

$$\ln(e^x e^y e^z) = 0$$

Gauname naują sąryšio sąlygą:

```
> simplify(%,ln,symbolic);
```

$$x + y + z = 0$$

Toliau įvedame kairiąją nelygybės pusę ir pakeičiame kintamuosius:

```
> R:=(a-1+1/b)*(b-1+1/c)*(c-1+1/a);
```

$$R := \left(a - 1 + \tfrac{1}{b}\right)\left(b - 1 + \tfrac{1}{c}\right)\left(c - 1 + \tfrac{1}{a}\right)$$

```
> simplify(subs(T,z=-x-y,R));
```

$$(e^x - 1 + e^{(-y)})(e^y - 1 + e^{(x+y)})(e^{(-x-y)} - 1 + e^{(-x)})$$

Apibrėžiame funkciją, kurios maksimumą reikia rasti:

```
f:=unapply(%,(x,y));
```

$$f := (x, y) \to (e^x - 1 + e^{(-y)})(e^y - 1 + e^{(x+y)})(e^{(-x-y)} - 1 + e^{(-x)})$$

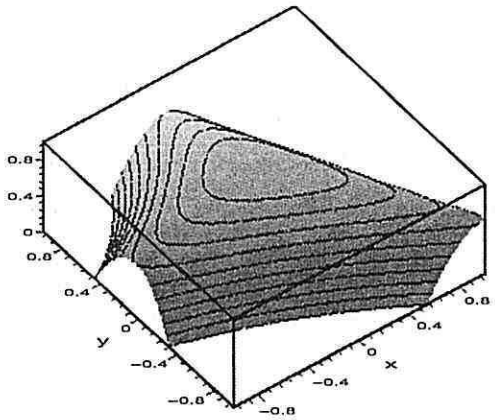
Randame kritinius taškus (galima naudoti ir komandą `fsolve`)

```
> _EnvExplicit:=true:
> solve({D[1](f)(x,y)=0,D[2](f)(x,y)=0},{x,y});
```

Gavome vieną kritinį tašką $M(0; 0)$. Kitas šaknis atmetame, nes jos yra kompleksinės. Toliau tiriame tą kritinį tašką. Iš pradžių nubraižome funkcijos $f(x, y)$ grafiką (2 pav.) taško $M(0; 0)$ aplinkoje.

$\alpha + \omega$

```
> plot3d(f(x,y),x=-1..1,y=-1..1,axes=boxed,style=PATCHCONTOUR,
view=[-1..1,-1..1,0..1],orientation=[-60,30],numpoints=2000);
Plotting error, empty plot
```



2 pav.

Kompiuterio ekrane brėžinį galima pavartyti ir apžiūrėti iš visų pusių, pakeisti braižymo stilių. Matome, kad $M(0;0)$ — maksimumo taškas. Apskaičiuojame funkcijos reikšmę šiame taške:

```
> f(0,0);
```

$$1$$

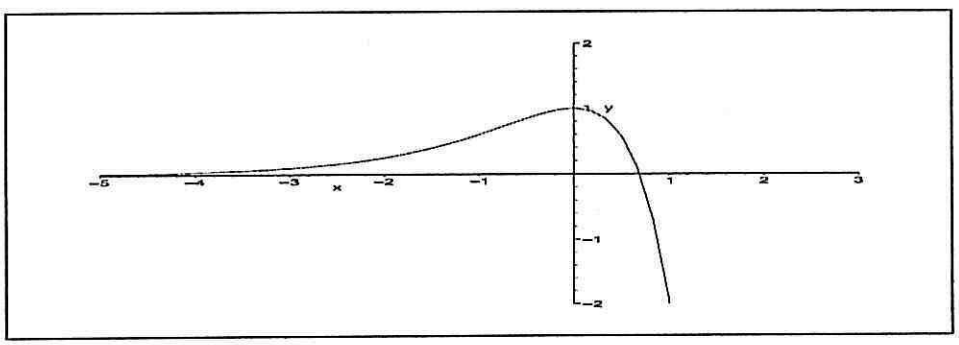
Fiksavus vieną kintamąjį, $f(x, y)$ galima tirti kaip vieno kintamojo funkciją naudojantis visa žinoma teorija. Pavyzdžiui, kai $y = 0$, turime

```
> simplify(expand(f(x,0)));
```

$$2e^{x} - e^{(2x)}$$

ir braižome grafiką (3 pav.):

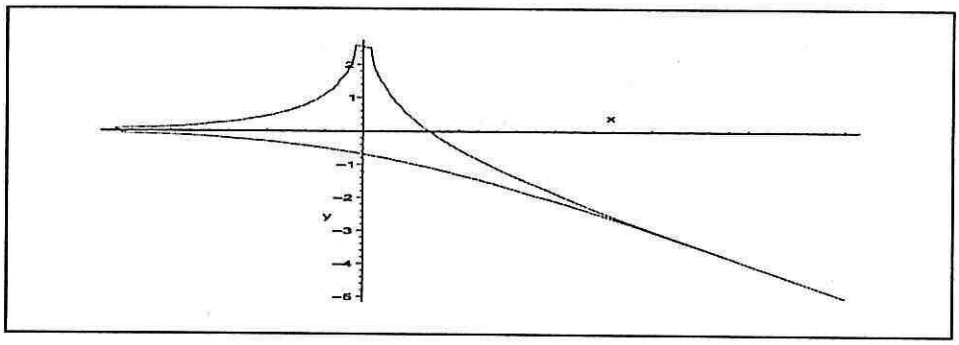
```
> plot(f(x,0),x=-5..3,y=-2..2);
```



3 pav.

Nubraižome kreivę (4 pav.), kuria funkcijos grafikas kertasi su plokštuma $z = 0$. Gauname kreivinį trikampį, kurio viduje funkcija yra teigiama, o jo išorėje — neigiama. Maksimumas yra įgyjamas koordinačių pradžioje ir jo reikšmė lygi 1.

```
> plots[implicitplot](f(x,y)=0,x=-5..5,y=-5..5,numpoints=4000);
```



4 pav.

Yra ir kitų funkcijos f grafinio ir skaitinio tyrimo galimybių. Kviečiame skaitytojus jas pabandyti rasti.

Toliau taikome ekstremumo nustatymo taisyklę. Tam kritiniame taške $M(0;0)$ apskaičiuojame antrąsias išvestines $A = f_{xx}$, $B = f_{yy}$, $C = f_{xy}$ ir determinantą $\Delta = AB - C^2$:

```
> A:=D[1,1](f)(0,0);
```

$$A := -2$$

```
> B:=D[2,2](f)(0,0);
```

$$B := -2$$

```
> C:=D[1,2](f)(0,0);
```

$$C := -1$$

```
> Delta:=A*B-C^2;
```

$$\Delta := 3$$

Matome, kad $\Delta > 0$ ir $A < 0$. Todėl $M(0;0)$ yra maksimumo taškas, maksimumo reikšmė lygi 1. Grįžkime prie senų kintamųjų. Maksimumas įgyjamas taške

```
> simplify(subs(z=-x-y,x=0,y=0,T));
```

$$\{c = 1, b = 1, a = 1\}$$

Maksimumo reikšmė lygi

```
> subs(%,R);
```

$$1$$

Ji neviršija nelygybės dešiniosios pusės. Todėl duotoji nelygybė įrodyta.

3 uždavinys ([3]). *Skaičių korys yra taisyklingo šešiakampio formos, kraštinės ilgis yra sveikasis skaičius iš intervalo* [1; 99], *akutėse įrašyti sveikieji skaičiai yra iš intervalo* [0; 99]. *Kelias prasideda kurioje nors viršutinės eilutės korio akutėje ir baigiasi kurioje nors apatinės eilutės korio akutėje. Iš akutės kelias gali eiti tik įstrižai žemyn į kairę arba įstrižai žemyn į dešinę. Sudarant kelią, vieną kartą leidžiama vienos horizontalios korio eilutės du skaičius sukeisti vietomis. Parašykite programą, kuri paskaičiuotų, kokia gali būti didžiausia kelią sudarančių akučių skaičių suma.*

```
> restart;
> with(plots,textplot,display):
> with(geometry,point,RegularPolygon,rotation):
> with(networks):
```

Rasime ne tik sumą, bet ir kelią, kurį pavaizduosime grafiškai. Įvesime pavyzdžio iš [3] duomenis (5 pav.). Duomenis pakeitus kitais, čia pateikiama programa turi veikti. Sprendimo laikas gali gerokai padidėti, jei didinsite kraštinės ilgį. Kai kraštinių ilgiai yra lygūs 3, kompiuteris su 600 Mz Pentium III procesoriumi šį uždavinį išsprendžia per 10 sekundžių. Jei nereikia rasti kelio ir jo braižyti, tai galima programuoti pagal greitesnį algoritmą, nurodytą [3].

Korio kraštinės ilgis:

```
> n:=3;
```

$$n := 3$$

Skaičiai akutėse:

```
> T:=[[1,2,3],[3,2,2,1],[4,2,8,0,3],[5,3,1,2],[3,1,4]];
```

$$T := [[1, 2, 3], [3, 2, 2, 1], [4, 2, 8, 0, 3], [5, 3, 1, 2], [3, 1, 4]]$$

```
> Ts:=map(op,T);
```

$$Ts := [1, 2, 3, 3, 2, 2, 1, 4, 2, 8, 0, 3, 5, 3, 1, 2, 3, 1, 4]$$

Akučių skaičius:

```
> m:=nops(%);
```

$$m := 19$$

Akutes žymėsime aij; čia i — eilutės numeris, j — akutės eilutėje numeris (6 pav.). Korio eilutės:

```
> E:=seq([seq(a||i||j,j=1..nops(T[i]))],i=1..2*n-1);
```

$E := [a11, a12, a13]$, $[a21, a22, a23, a24]$, $[a31, a32, a33, a34, a35]$, $[a41, a42, a43, a44]$, $[a51, a52, a53]$

Visos akutės:

```
> S:=seq(op(E[k]),k=1..2*n-1);
```

$S := a11, a12, a13, a21, a22, a23, a24, a31, a32, a33, a34, a35, a41, a42, a43, a44, a51, a52, a53$

Eilutėse randame akutes, kuriose įrašyti didžiausi eilučių skaičiai:

```
> Em:=NULL:
> for k to 2*n-1 do
> member(max(op(T[k])),T[k],'p');
> Em:=Em,a||k||p;
> end do:  Em;
```

$$a13, a21, a33, a41, a53$$

Funkcija akučių centrų koordinatėms apskaičiuoti:

```
> F:=(i,j)->if i<=n then [sqrt(3)*(j-n)+sqrt(3)*(n-i)/2,(n-i)*
3/2] else [sqrt(3)*(j-n)-sqrt(3)*(n-i)/2,(n-i)*3/2] end if:
```

Randame korio akučių centrų koordinates:

```
> C:=NULL: for i to 2*n-1 do for j to nops(T[i]) do C:=C,F(i,j);
end do; end do; C;
```

$[-\sqrt{3}, 3]$, $[0, 3]$, $[\sqrt{3}, 3]$, $[-\frac{3}{2}, \sqrt{3}, \frac{3}{2}]$, $[-\frac{1}{2}\sqrt{3}, \frac{3}{2}]$, $[\frac{1}{2}\sqrt{3}, \frac{3}{2}]$, $[\frac{3}{2}\sqrt{3}, \frac{3}{2}]$, $[-2\sqrt{3}, 0]$,

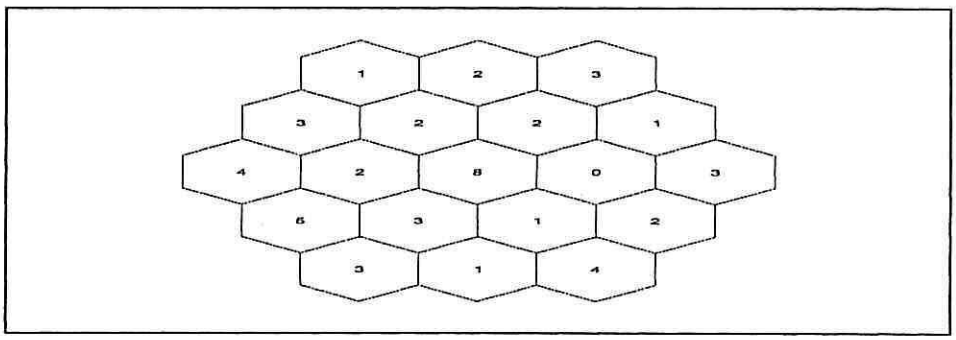
$[-\sqrt{3}, 0]$, $[0, 0]$, $[\sqrt{3}, 0]$, $[2\sqrt{3}, 0]$, $[-\frac{3}{2}\sqrt{3}, \frac{-3}{2}]$, $[-\frac{1}{2}\sqrt{3}, \frac{-3}{2}]$, $[\frac{1}{2}\sqrt{3}, \frac{-3}{2}]$, $[\frac{3}{2}\sqrt{3}, \frac{-3}{2}]$, $[-\sqrt{3}, -3]$, $[0, -3]$, $[\sqrt{3}, -3]$

Įvedame korio akutes:

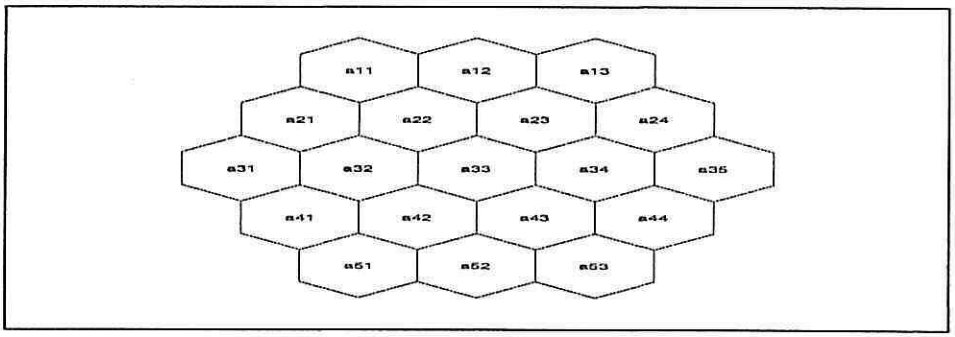
```
> seq(RegularPolygon(d||k,6,point(o||k,C[k]),1),k=1..m):
> akutes:=seq(rotation(A||k,d||k, Pi/6,'clockwise', o||k),k=1..m):
```

Jas nubraižome (5 pav. ir 6 pav.):

```
> K:=geometry[draw]([akutes],axes=none):
> display(K,seq(textplot([op(C[k]),Ts[k]]),k=1..m));br:=%:
> display(K,seq(textplot([op(C[k]),S[k]]),k=1..m));
```



5 pav.



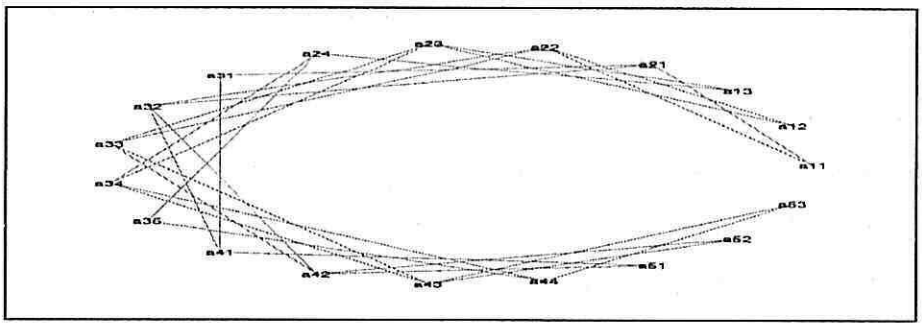
6 pav.

Optimaliam keliui rasti pasinaudosime grafų teorijos ***networks*** paketu.

Apibrėžiame orientuotąjį grafą G (7 pav.):

```
> new(G):
> for i to m do addvertex(S[i],weights=Ts[i],G); end do:
> for i to n-1 do for j to n+i-1 do connect({a||i||j},{a||(i+1)||j,
a||(i+1)||(j+1)},'directed',G); end do; end do;
> for i from n to 2*n-2 do for j to 3*n-i-2 do connect({a||i||j,
a||i||(j+1)}, {a|| (i+1)||j},'directed',G); end do; end do;
> draw(G);
```

$\alpha + \omega$



7 pav.

Iš pradžių sudarome procedūrą optimaliam keliui rasti, kai neleidžiama sukeisti:

```
> opt_kelias:=proc(G) local suma,i,j,tkelias,kelias,k; suma:=0:
for i to n do for j to n do flow(G,(a||1||i), (a||(2*n-1)||j),
'eset','comp','maxflow'=1); tkelias:=map(op,eset);
seq(vweight(%[k],G),k=1..2*n-1); sum(%[k],k=1..2*n-1); if suma<%
then suma:=%; kelias:=select(has,[S], tkelias); end if; end do;
end do; RETURN (suma,kelias); end proc:
```

Jei nebūtų leidžiama sukeisti, tai didžiausia suma ir optimalus kelias būtų:

```
> opt_kelias(G);
```

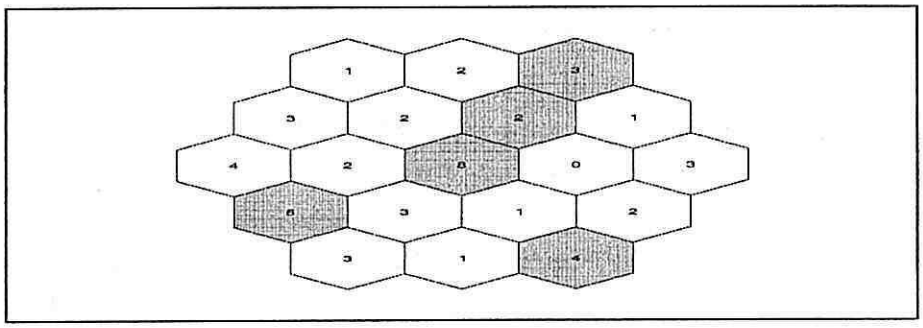
$$19,\ [a13, a23, a33, a42, a51]$$

Toliau kiekvieną eilutę pakeičiame didžiausiu eilutės skaičiumi ir kiekvienu atveju randame optimalų kelią. Iš visų kelių išsirenkame tą, kurio akučių suma yra didžiausia:

```
> a_suma:=0:
> for k to 2*n-1 do H:=duplicate(G): for i to nops(T[k]) do
addvertex(E[k][i],weights=max(op(T[k])),H); end do; sp:=
opt_kelias(H); if %[1]>a_suma then a_suma:=sp[1]; sprend:=
subs(sp[2][k]=Em[k],sp[2]); end if; end do:
```

Atrenkame rasto kelio akutes ir braižome (8 pav.):

```
> sp:=NULL: for k to 2*n-1 do member(sprend[k],[S],'p');
sp:=sp,akutes[p]; end do:
> geometry[draw]([sp],filled=true,color=grey):
display(%,br,axes=none);
```



8 pav.

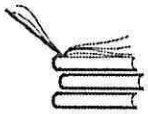
Gauname atsakymą:

```
> 'Suma'=a_suma;
```

$$Suma = 22$$

```
> 'Kelias'=sprend;
```

$$Kelias = [a13, a23, a33, a41, a53]$$



1. G. Alkauskas, Komutuojantys daugianariai, *Alfa plius omega*, **1**, 53–56, 2001.
2. J. Mačys, Ar įkandami pasaulinės olimpiados uždaviniai? *Alfa plius omega*, **1**, 75–80, 2001.
3. J. Skūpienė, Korys, *Alfa plius omega*, **1**, 81–83, 2001.
4. A. Domarkas, *Matematikos praktikumas su Maple*, http://ieva.maf.vu.lt/home/aleksas
5. V. P. Djakonov, Kompiuterinė matematika (rusų k.), Maskva, 2001, 1296 p.

Atrodo, kad matematikai pamiršo, jog du tūkstančius metų Euklido geometrija buvo griežtumo etalonas. Net didieji matematikai nematė jos trūkumų. O realiųjų skaičių pagrindai buvo sukurti tik antroje devyniolikto amžiaus pusėje. Nei Oileris, nei Gausas nebūtų galėjęs apibrėžti realiųjų skaičių, kažin ar juos apskritai būtų džiuginę varginančios šio apibrėžimo detalės. Tačiau abu jie itin gerai suprato matematiką ir gavo neprilygstamų rezultatų. Griežti įrodymai nėra tokie svarbūs kaip įrodymas, kad tai, ko mokome, yra svarbu. Daugeliui dėstytojų, užuot rūpinusis, kad dėstymas būtų pakankamai griežtas, verta susirūpinti tuo, kaip sukurti ryškų intuityvų vaizdinį... Bendrasis principas yra toks: griežtumo lygis turi atitikti matematinę studentų brandą, o ne matematikos brandą.

Morris Kline, *Mathematics: A Cultural Approach*, Addison-Wesley, 1966, p. vii.