ÚJ SZÉCHENYI TERV

Matematikai problémák vizsgálata a Maple programcsomag segítségével

Tengely Szabolcs

tengely@science.unideb.hu

http://www.math.unideb.hu/~tengely

Áttekintés

Feszítőfák keresése - Kruskal

Adott $G(V, E)$ gráf és egy $s : E \longrightarrow \mathbb{R}_+$ súlyfüggvény.
Keresünk olyan feszítőfát, amelyben az élek súlya minimális.
Kruskal algoritmus: rendezzük az éleket növekvő sorrendbe a súlyok szerint, addig választunk be éleket, amíg fát nem kapunk.
Delete-reverse algoritmus: csökkenő sorrendbe rendezzük az éleket és addig törlünk, amíg összefüggő gráf marad.

Feszítőfák keresése - Kruskal

```
> FeszitoFa1:=proc(G)
> local g0,f,eG,l; g0:=Graph(NumberOfVertices(G),'weighted');
> f:=(x,y)->if x[2]<y[2] then true; else false; end if;
> eG:=sort([op(Edges(G,weights))],f);
> l:=1; while not IsTree(g0) do
> if IsForest(AddEdge(g0,eG[l][1],inplace=false)) then g0:=AddEdge
  (g0,eG[l]); end if;
> l:=l+1;
> end do;
> g0;
> end proc;
```

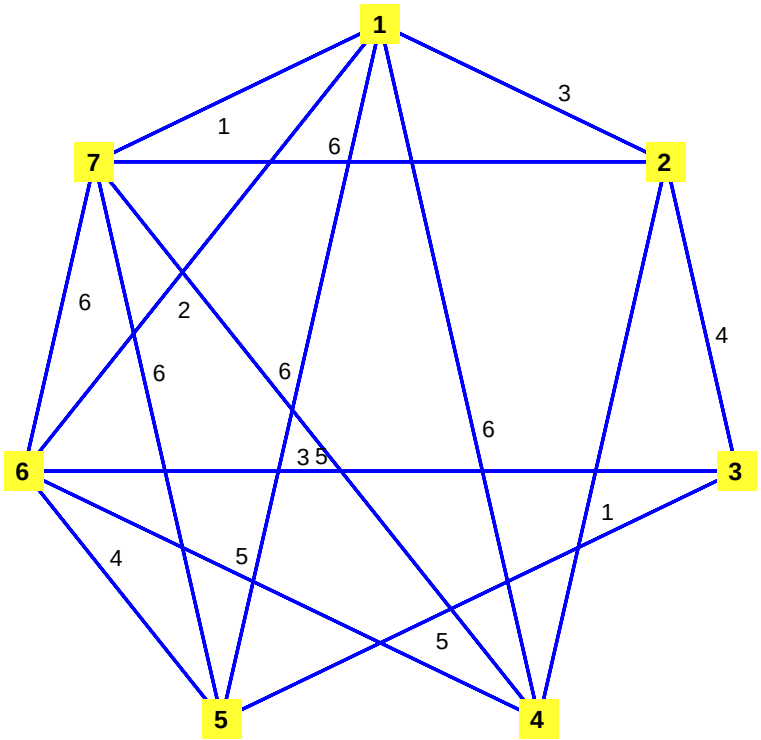
```
FeszitoFa1 := proc(G)                                                    (1)
    local g0, f, eG, l;
    g0 := GraphTheory:-Graph(GraphTheory:-NumberOfVertices(G), 'weighted');
    f := (x, y) → if x[2] < y[2] then true else false end if;
    eG := sort([op(GraphTheory:-Edges(G, weights))], f);
    l := 1;
    while not GraphTheory:-IsTree(g0) do
        if GraphTheory:-IsForest(GraphTheory:-AddEdge(g0, eG[l][1], inplace = false)) then
            g0 := GraphTheory:-AddEdge(g0, eG[l])
        end if;
        l := l + 1
    end do;
    g0
end proc
```

```
> with(GraphTheory):
> with(RandomGraphs):
> Graf:=RandomGraph(7,15,connected, weights=1..6);
```

```
Graf := Graph 1: an undirected weighted graph with 7 vertices and 15 edge(s)   (2)
```

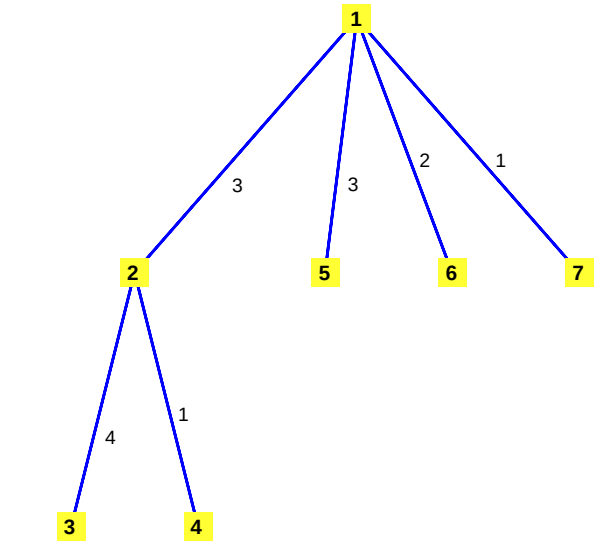
```
> DrawGraph(Graf);
```

Feszítőfák keresése - Kruskal




```
> DrawGraph(FeszitoFa1(Graf));
```

Feszítőfák keresése - Reverse-Delete




```
> FeszitoFa2:=proc(G)
> local f,eG,H,l; f:=(x,y)->if x[2]>y[2] then true; else false; end
  if;
> eG:=sort([op(Edges(G,weights))],f);
> H:=G; l:=1;
> while not IsTree(H) do
> if IsConnected(DeleteEdge(H,eG[l][1],inplace=false)) then
> DeleteEdge(H,eG[l]); end if;
> l:=l+1; end do;
> H;
> end proc;
```

$$FeszitoFa2 := \mathbf{proc}(G) \qquad (3)$$

$$\mathbf{local}\ f, eG, H, l;$$

$$f := (x, y) \rightarrow \mathbf{if}\ y[2] < x[2]\ \mathbf{then}\ true\ \mathbf{else}\ false\ \mathbf{end\ if};$$

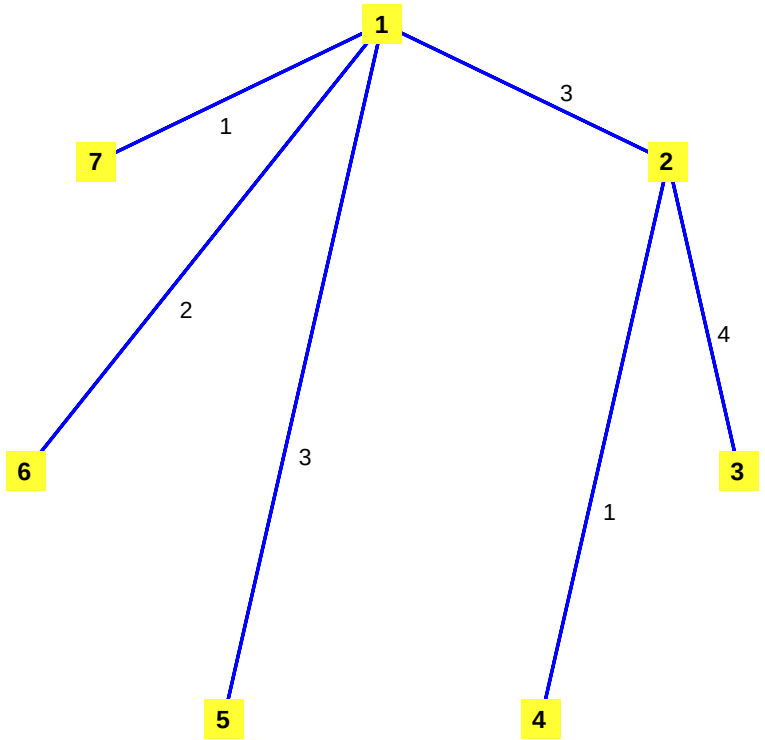
$$eG := sort([op(GraphTheory:-Edges(G, weights))], f);$$

$$H := G;$$

Feszítőfák keresése - Reverse-Delete

```
    l := 1;
    while not GraphTheory:-IsTree(H) do
        if GraphTheory:-IsConnected(GraphTheory:-DeleteEdge(H, eG[l][1], inplace
        =false)) then
            GraphTheory:-DeleteEdge(H, eG[l])
        end if;
        l := l+1
    end do;
    H
end proc
> DrawGraph(FeszitoFa2(Graf));
```

Feszítőfák keresése - Reverse-Delete



Sakktábla bejárása

$m \times n$-es sakktábla bejárása

Adott $m \times n$-es sakktábla esetén egy mezőről indulva lóugrással járjuk be a táblát úgy, hogy minden mezőt pontosan egyszer érintünk és a kiindulópontba jutunk vissza.
Gráfelméletre lehet visszavezetni. Csúcsok: $[a_1, a_2], [b_1, b_2], \ldots$

$$|a_1 - b_1||a_2 - b_2| = 2.$$

Sakktábla bejárása

```
> with(GraphTheory):
> csucs:=(m,n)->{seq(seq([a,b],a=1..m),b=1..n)};
```

$$csucs := (m, n) \rightarrow \{seq(seq([a, b], a = 1..m), b = 1..n)\}$$

```
> f:=x->if abs(x[1,1]-x[2,1])*abs(x[1,2]-x[2,2])=2 then true; else false
  end if;
```

$$f := x \rightarrow \textbf{if } |x_{1,1} - x_{2,1}| \, |x_{1,2} - x_{2,2}| = 2 \textbf{ then } true \textbf{ else } false \textbf{ end if}$$

```
> A:=(m,n)->select(f,{seq(seq([u,v],u=csucs(m,n)),v=csucs(m,n))});
```

$$A := (m, n) \rightarrow select(f, \{seq(seq([u, v], u = csucs(m, n)), v = csucs(m, n))\})$$

```
> B:=(m,n)->{seq({convert(k[1],string),convert(k[2],string)},k=A(m,n))};
```

$$B := (m, n) \rightarrow \{seq(\{convert(k_1, string), convert(k_2, string)\}, k = A(m, n))\}$$

```
> Gsakk:=(m,n)->Graph(B(m,n));
```

$$Gsakk := (m, n) \rightarrow GraphTheory:-Graph(B(m, n))$$

```
> sakkgraf:=Gsakk(6,6);
```

$$sakkgraf := Graph\ 1:\ an\ undirected\ unweighted\ graph\ with\ 36\ vertices\ and\ 80\ edge(s)$$

```
> IsHamiltonian(sakkgraf,'s');
```

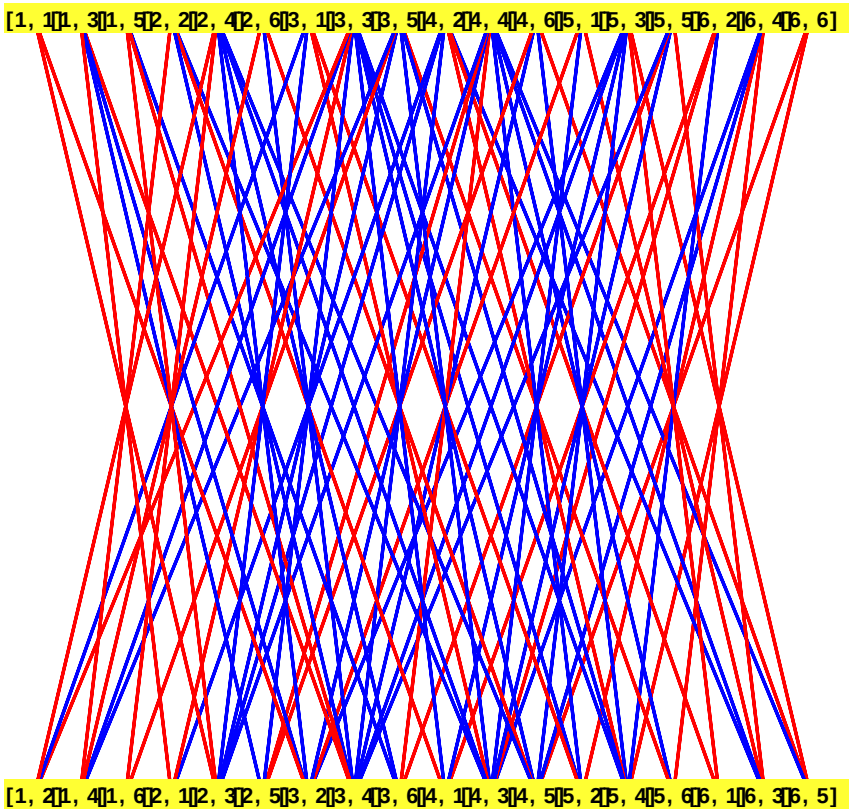
$$true$$

```
> s;
```

["[1, 1]", "[2, 3]", "[1, 5]", "[3, 4]", "[1, 3]", "[2, 1]", "[3, 3]", "[1, 2]", "[2, 4]", "[1, 6]", "[3, 5]", "[5, 6]", "[6, 4]", "[5, 2]", "[3, 1]", "[4, 3]", "[6, 2]", "[4, 1]", "[2, 2]", "[1, 4]", "[2, 6]", "[4, 5]", "[6, 6]", "[5, 4]", "[4, 2]", "[6, 1]", "[5, 3]", "[6, 5]", "[4, 6]", "[2, 5]", "[4, 4]", "[3, 6]", "[5, 5]", "[6, 3]", "[5, 1]", "[3, 2]", "[1, 1]"]

```
> HighlightTrail(sakkgraf,s,red);
> DrawGraph(sakkgraf);
```

Sakktábla bejárása



Gráfok színezése - algebrai út

Csúcsok színezése

k-színezés:

$$f(x) = x(x-1)(x-2)\cdots(x-k+1)$$

véges test feletti polinom.

$$f(x_1) = 0, \quad f(x_2) = 0, \ldots, f(x_n) = 0$$

garantálja, hogy minden csúcsot kiszínezünk.

Gráfok színezése - algebrai út

Csúcsok színezése

$$f(x_i) - f(x_j)$$

osztható $x_i - x_j$-vel.

$$g(x_i, x_j) = \frac{f(x_i) - f(x_j)}{x_i - x_j}.$$

Minden $x_i x_j \in E$ esetében $g(x_i, x_j) = 0$.

Gráfok színezése - algebrai út

```
> szin:=proc(G,k)
> local f,g,n,rendszer, el,T,t; f:=x->expand(product(x-a,a=0..k-1));
> g:=(x,y)->simplify((f(x)-f(y))/(x-y));
> n:=NumberOfVertices(G);
> rendszer:={x[1]} union {seq(f(x[j]),j=2..n)};
> el:=Edges(G);
> for T in el do t:=op(T);
> rendszer:=rendszer union {g(x[t[1]],x[t[2]])};
> end do;
> rendszer;
> end proc;
```

```
szin := proc(G, k)
    local f, g, n, rendszer, el, T, t;
    f := x → expand(product(x − a, a = 0 ..k − 1));
    g := (x, y) → simplify((f(x) − f(y))/(x − y));
    n := GraphTheory:-NumberOfVertices(G);
    rendszer := union({x[1]}, {seq(f(x[j]), j = 2 ..n)});
    el := GraphTheory:-Edges(G);
    for T in el do t := op(T); rendszer := union(rendszer, {g(x[t[1]], x[t[2]])}) end do;
    rendszer
end proc
```

```
> with(GraphTheory): K3:=CompleteGraph(3);
```

$$K3 := \text{Graph 2: an undirected unweighted graph with 3 vertices and 3 edge(s)}$$

```
> szin(K3,2);
```

$$\{x_1, x_2^2 - x_2, x_3^2 - x_3, x_1 + x_2 - 1, x_1 + x_3 - 1, x_2 + x_3 - 1\}$$

```
> [msolve(szin(K3,2),3)];
```

$$[\,]$$

```
> [msolve(szin(K3,3),3)];
```

$$[\{x_1 = 0, x_2 = 1, x_3 = 2\}, \{x_1 = 0, x_2 = 2, x_3 = 1\}]$$

```
> with(SpecialGraphs): P:=PetersenGraph();
```

$$P := \text{Graph 3: an undirected unweighted graph with 10 vertices and 15 edge(s)}$$

```
> A:=[msolve(szin(P,3),5)];
```

$$A := [\{x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 2, x_6 = 1, x_7 = 2, x_8 = 0, x_9 = 2, x_{10} = 0\}, \{x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 2, x_6 = 2, x_7 = 1, x_8 = 0, x_9 = 2, x_{10} = 0\}, \{x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 2, x_5 = 1, x_6 = 1, x_7 = 2, x_8 = 0, x_9 = 2, x_{10} = 0\}, \{x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 2, x_5 = 1, x_6 = 2, x_7 = 1, x_8 = 0, x_9 = 2, x_{10} = 0\}, \{x_1 = 0, x_2 = 2, x_3 = 0, x_4 = 1, x_5 = 2, x_6 = 1, x_7 = 2, x_8 = 0, x_9 = 1, x_{10} = 0\}, \{x_1 = 0, x_2 = 2, x_3 = 0, x_4 = 1, x_5 = 2, x_6 = 2, x_7 = 1, x_8 = 0, x_9 = 1, x_{10} = 0\}, \{x_1 = 0, x_2 = 2, x_3 = 0, x_4 = 2, x_5 = 1, x_6 = 1, x_7 = 2, x_8 = 0, x_9 = 1, x_{10} = 0\}, \{x_1 = 0, x_2 = 2, x_3 = 0, x_4 = 2, x_5 = 1, x_6 = 2, x_7 = 1, x_8 = 0, x_9 = 1, x_{10} = 0\}, \{x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 0, x_5 = 1, x_6 = 1, x_7 = 0, x_8 = 2, x_9 = 0, x_{10} = 2\}, \{x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 0, x_5 = 1, x_6 = 2, x_7 = 0, x_8 = 2, x_9 = 0, x_{10} = 1\}, \{x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 0, x_5 = 2, x_6 = 1, x_7 = 0, x_8 = 1, x_9 = 0, x_{10} = 2\}, \{x_1 = 0, x_2 = 1, x_3 = 2, x_4 = 0, x_5 = 2, x_6 = 2, x_7 = 0, x_8 = 1, x_9 = 0, x_{10} = 1\}, \{x_1 = 0, x_2 = 2, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 1, x_7 = 0, x_8 = 2, x_9 = 0, x_{10} = 2\}, \{x_1 = 0, x_2 = 2, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 2, x_7 = 0, x_8 = 2, x_9 = 0, x_{10} = 1\}, \{x_1 = 0, x_2 = 2, x_3 = 1, x_4 = 0, x_5 = 2,$$

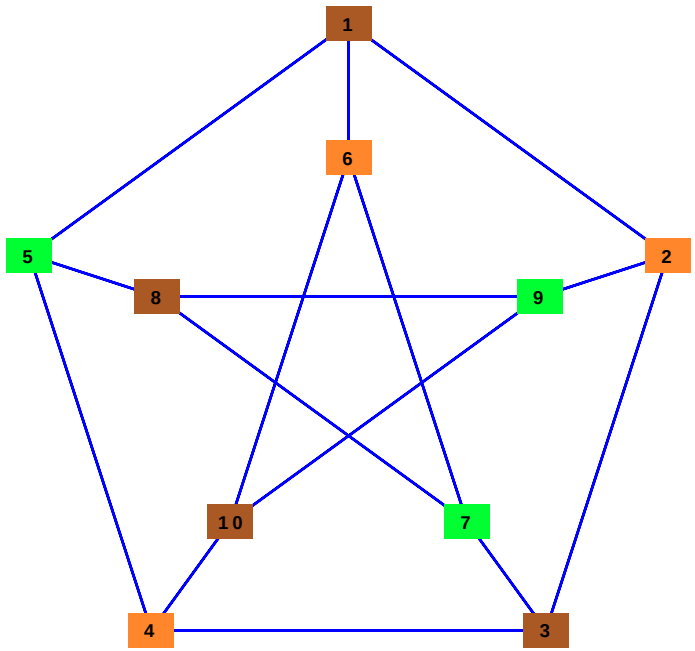
Gráfok színezése - algebrai út

$$x_6 = 1, x_7 = 0, x_8 = 1, x_9 = 0, x_{10} = 2\}, \{x_1 = 0, x_2 = 2, x_3 = 1, x_4 = 0, x_5 = 2, x_6 = 2, x_7 = 0, x_8 = 1, x_9 = 0, x_{10} = 1\}]$$

```
> a1:=[seq(op(A[1][k])[2],k=1..10)];
```

$$a1 := [0, 1, 0, 1, 2, 1, 2, 0, 2, 0]$$

```
> for s from 1 to 10 do HighlightVertex(P,s,COLOR(RGB,1/(1.5-a1[s]),1/(2.9
  -a1[s]),1/(7.1-a1[s])));
> end do;
> DrawGraph(P);
```



Mátrixok hatványai sajátvektorokkal

Diagonalizálás

A egy mátrix, P a sajátvektorokból álló mátrix:

$$P^{-1}AP$$

diagonális. Ezt felhasználva meghatározhatjuk az A^n mátrixot zárt alakban.

Rekurzív sorozatok

Fibonacci sorozat

$F_0 = 0, F_1 = 1$ és $F_n = F_{n-1} + F_{n-2}$, legyen

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

ekkor $A^n \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$.

Inhomogén rekurzív egyenletrendszer

Inhomogén rekurzió zárt alakjának meghatározása

$x_{n+1} = ax_n + by_n + c$

$y_{n+1} = dx_n + ey_n + f$

adott x_0, y_0 kezdőértékek mellett.

a -2
b 3
c 2
d -1
e 2
f -2
x0 3
y0 2

Az A mátrix: $\begin{pmatrix} -2 & 3 \\ -1 & 2 \end{pmatrix}$
A B vektor: $(2, -2)$
Az X_0 vektor: $(3, 2)$
Az A mátrix sajátértékei: -1 és 1
sajátérték: -1, hozzá tartozó sajátvektor: $\left(1, \frac{1}{3}\right)$
sajátérték: 1, hozzá tartozó sajátvektor: $(1, 1)$
A sajátvektorokból álló P mátrix: $\begin{pmatrix} 1 & 1 \\ \frac{1}{3} & 1 \end{pmatrix}$
P inverze: $\begin{pmatrix} \frac{3}{2} & -\frac{3}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{pmatrix}$
Ekkor a $P^{-1}AP$ mátrix diagonális: $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
Innen A^n zárt alakja könnyen meghatározható: $A^n = PD^nP^{-1}$
Az A mátrix n-edik hatványa: $\begin{pmatrix} \frac{3}{2}(-1)^n - \frac{1}{2} & -\frac{3}{2}(-1)^n + \frac{3}{2} \\ \frac{1}{2}(-1)^n - \frac{1}{2} & -\frac{1}{2}(-1)^n + \frac{3}{2} \end{pmatrix}$
Az $\begin{pmatrix} x_n \\ y_n \end{pmatrix} = A^n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + (A^{n-1} + A^{n-2} + \ldots + A + E)B$ összefüggés alapján adódik a formula x_n-re és y_n-re:

$x_n = -\frac{3}{2}(-1)^n - 4n + \frac{9}{2}$

$y_n = -\frac{1}{2}(-1)^n - 4n + \frac{5}{2}$

Mátrixok hatványai sajátvektorokkal

```
> with(LinearAlgebra):
> matrixN:=proc(A)
> local u,v,Diag; u,v:=Eigenvectors(A);
> Diag:=Matrix([[u[1]^n,0],[0,u[2]^n]]);
> v.Diag.v^(-1);
> end proc;
```

$$matrixN := \mathbf{proc}(A)\ \mathbf{local}\ u, v, Diag;\ u, v := LinearAlgebra{:}\text{-}Eigenvectors(A);\ Diag := Matrix([[u[1]^n, 0], [0, u[2]^n]]);\ \cdot(v, Diag, 1/v)\ \mathbf{end\ proc}$$

```
> matrixN(Matrix([[4,-3],[1,0]]));
```

$$\begin{bmatrix} -\frac{1}{2}+\frac{3\,3^n}{2} & \frac{3}{2}-\frac{3\,3^n}{2} \\ -\frac{1}{2}+\frac{3^n}{2} & \frac{3}{2}-\frac{3^n}{2} \end{bmatrix}$$

```
> Fib:=(matrixN(Matrix([[1,1],[1,0]])).Vector([1,0]))[2];
```

$$Fib := \frac{\left(\frac{\sqrt{5}}{2}+\frac{1}{2}\right)^n \sqrt{5}\,(\sqrt{5}-1)\,(\sqrt{5}+1)}{20} - \frac{\left(\frac{1}{2}-\frac{\sqrt{5}}{2}\right)^n \sqrt{5}\,(\sqrt{5}-1)\,(\sqrt{5}+1)}{20}$$

```
> simplify(Fib);
```

$$\frac{\sqrt{5}\left(\left(\frac{\sqrt{5}}{2}+\frac{1}{2}\right)^n-\left(\frac{1}{2}-\frac{\sqrt{5}}{2}\right)^n\right)}{5}$$

```
> seq(evala(subs(n=k,Fib)),k=0..10);
```

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55$$

```
> A:=Matrix([[2,1],[1,2]]);
```

$$A := \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

```
> U1:=subs(t=n,sum(matrixN(A),n=0..t-1)).Vector([1,1]);
```

$$U1 := \begin{bmatrix} -\frac{1}{2}+\frac{3^n}{2} \\ -\frac{1}{2}+\frac{3^n}{2} \end{bmatrix}$$

```
> U2:=matrixN(A).Vector([2,3]);
```

$$U2 := \begin{bmatrix} -\frac{1}{2}+\frac{5\,3^n}{2} \\ \frac{1}{2}+\frac{5\,3^n}{2} \end{bmatrix}$$

```
> U1+U2;
```

$$\begin{bmatrix} -1+3\,3^n \\ 3\,3^n \end{bmatrix}$$

Berlekamp algoritmus

Polinomok $\mathbb{F}_q[X]$ felett

$f(X) = \sum_{k=0}^{n} a_i X^i$ ekkor a kis Fermat-tétel miatt

$$f(X)^q = \sum_{k=0}^{n} a_i X^{qi}.$$

Az $X^q - X$ polinom faktorizációja egyszerű. Egy F polinom faktorizációjához keresunk olyan $f(X)$-et, amelyre

$$f(X) \equiv f(X)^q \pmod{F}.$$

Berlekamp algoritmus

```
> P:=x^5+10*x^4+25*x^2+28;
```
$$P := x^5 + 10\,x^4 + 25\,x^2 + 28$$
```
> h:=add(s,s=[seq(a[i]*x^i,i=0..4)]);
```
$$h := a_0 + a_1\,x + a_2\,x^2 + a_3\,x^3 + a_4\,x^4$$
```
> h31:=add(s,s=[seq(a[i]*x^(31*i),i=0..4)]);
```
$$h31 := a_0 + a_1\,x^{31} + a_2\,x^{62} + a_3\,x^{93} + a_4\,x^{124}$$
```
> for k from 31 to 124 do h31:=subs(x^k=modpol(x^k,P,x,31),h31); end do:
  h31;
```
$$a_0 + a_1\,(8\,x^4 + 9\,x^3 + 9\,x^2 + 28\,x + 19) + a_2\,(26\,x^4 + x^3 + 7\,x^2 + 3\,x + 3) + a_3\,(7\,x^4 + 3\,x^3 + 4\,x^2 + 6\,x + 28) + a_4\,(24\,x^4 + 26\,x^3 + 22\,x^2 + 16\,x + 28)$$
```
> with(LinearAlgebra):
> sys:=[seq(coeff(h31,x,k)=coeff(h,x,k),k=0..4)];
```
$$sys := [a_0 + 19\,a_1 + 3\,a_2 + 28\,a_3 + 28\,a_4 = a_0, 28\,a_1 + 3\,a_2 + 6\,a_3 + 16\,a_4 = a_1, 9\,a_1 + 7\,a_2 + 4\,a_3 + 22\,a_4 = a_2, 9\,a_1 + a_2 + 3\,a_3 + 26\,a_4 = a_3, 8\,a_1 + 26\,a_2 + 7\,a_3 + 24\,a_4 = a_4]$$
```
> var:=[seq(a[k],k=0..4)];
```
$$var := [a_0, a_1, a_2, a_3, a_4]$$
```
> (A,b):=GenerateMatrix(sys,var);
```
$$A, b := \begin{bmatrix} 0 & 19 & 3 & 28 & 28 \\ 0 & 27 & 3 & 6 & 16 \\ 0 & 9 & 6 & 4 & 22 \\ 0 & 9 & 1 & 2 & 26 \\ 0 & 8 & 26 & 7 & 23 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
```
> with(LinearAlgebra[Modular]):
> A31:=Mod(31,A,integer[]);
```
$$A31 := \begin{bmatrix} 0 & 19 & 3 & 28 & 28 \\ 0 & 27 & 3 & 6 & 16 \\ 0 & 9 & 6 & 4 & 22 \\ 0 & 9 & 1 & 2 & 26 \\ 0 & 8 & 26 & 7 & 23 \end{bmatrix}$$
```
> N:=Basis(31,A31,row,false,column);
```
$$N := \left[\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 26 \\ 12 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 17 \\ 7 \\ 0 \\ 1 \end{bmatrix} \right]$$
```
> T1:=x^3+12*x^2+26*x;
```
$$T1 := x^3 + 12\,x^2 + 26\,x$$
```
> for k from 0 to 30 do PT:=Gcd(P,T1+k) mod 31; if PT<>1 then print(PT);
  end if; end do:
```

Berlekamp algoritmus

$$x^2 + 15\,x + 9$$

$$x + 1$$

$$x^2 + 25\,x + 10$$

```
> T2:=x^4+7*x^2+17*x;
```

$$T2 := x^4 + 7\,x^2 + 17\,x$$

```
> for k from 0 to 30 do PT:=Gcd(P,T2+k) mod 31; if PT<>1 then print(PT);
  end if; end do:
```

$$x + 1$$

$$x^2 + 25\,x + 10$$

$$x^2 + 15\,x + 9$$

Rácsok

Adott $b_1, b_2, \ldots, b_k$ vektorok $\mathbb{R}^n$-ben,

$$\Lambda = \{z_1 b_1 + z_2 b_2 + \ldots + z_k b_k : z_i \in \mathbb{Z}\}$$

egy rács.

Adott rácsot több vektorrendszer is meg tud határozni. Rövid vektorokból álló rendszerek a gyakorlatban hasznosak (titkosítások). Az LLL-algoritmus segítségével találunk rövid vektorokat. Bemutatunk néhány érdekes alkalmazást.

Polinomok gyökei

```
> f:=expand(cos(16*x));
```

$$f := 32768\cos(x)^{16} - 131072\cos(x)^{14} + 212992\cos(x)^{12} - 180224\cos(x)^{10} + 84480\cos(x)^{8} - 21504\cos(x)^{6} + 2688\cos(x)^{4} - 128\cos(x)^{2} + 1$$

```
> cos(Pi);
```

$$-1$$

```
> F:=subs(cos(x)=X,f)+1;
```

$$F := 32768X^{16} - 131072X^{14} + 212992X^{12} - 180224X^{10} + 84480X^{8} - 21504X^{6} + 2688X^{4} - 128X^{2} + 2$$

```
> Fsol:=solve(F,X);
```

$$Fsol := \frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}, -\frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}, \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}, -\frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}, \frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}, -\frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}, \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}, -\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}$$

```
> evalf(Fsol[13]);
```

$$0.9807852805$$

```
> a:=evalf[25](cos(Pi/16));
```

$$a := 0.9807852804032304491261822$$

```
> with(LinearAlgebra):
> M:=Matrix(9,shape=identity);
```

$$M := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> M1:=<M|<seq(round(evalf[100](10^50*cos(Pi/16)^k)),k=0..8)>>;
```

Polinomok gyökei

$$M1 := \begin{bmatrix} 1&0&0&0&0&0&0&0&0&100000000000000000000000000000000000000000000000000 \\ 0&1&0&0&0&0&0&0&0&98078528040323044912618223613423903697393373089334 \\ 0&0&1&0&0&0&0&0&0&96193976625564337806409159469839414341120831293182 \\ 0&0&0&1&0&0&0&0&0&94345636337805914611433377150515571691509050116682 \\ 0&0&0&0&1&0&0&0&0&92532811390396181861419713996150027332181280514288 \\ 0&0&0&0&0&1&0&0&0&90754819366028955633741219146127825173794240997653 \\ 0&0&0&0&0&0&1&0&0&89010990959855238312675856242535366890020654926785 \\ 0&0&0&0&0&0&0&1&0&87300669727531030529873491813736155542421421228179 \\ 0&0&0&0&0&0&0&0&1&85623211838106333750179359857343053200094917034005 \end{bmatrix}$$

```
> with(IntegerRelations):
> T:=LLL(M1);
```

$T :=$

[[−1, 0, 32, 0, −160, 0, 256, 0, −128, 64],
[127695, −363515, 247964, −444872, −73655, 619509, −221822, 306020, −179583, 222866],
[−232341, 467650, 430323, −122787, −193639, −35791, 42783, −490737, 90598, −692681],
[−323635, 675132, 299256, 106890, −452280, 83728, −421829, 53579, −65423, 271165],
[240925, −48946, −87702, −241280, 504386, 328225, 101658, −356585, −496018, −90870],
[−59104, 663274, −355149, −588745, −212637, −169266, 29002, 553672, 171523, −127373],
[165139, −259929, −428390, 349529, −234744, 653417, −130316, −334610, 238762, 629101],
[435848, −638510, −379071, 623947, −212765, 429577, −161061, 149342, −249499, −190109],
[980750, 74502, −223188, −528380, −267779, −67159, −213928, 33515, 150976, 614698]]

```
> fx:=add(u,u=[seq(T[1,t]*x^(t-1),t=1..9)]);
```

$$fx := -1 + 32\,x^2 - 160\,x^4 + 256\,x^6 - 128\,x^8$$

```
> solve(fx,x);
```

$$\frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}, -\frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}, \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}, -\frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2},$$
$$\frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}, -\frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}, \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}, -\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}$$

Polinomok faktorizációja

```
> f:=expand((x^5+x^2+2014)*(x^6+x^3+x+2014));
```

$$f := x^{11} + 2x^8 + 2015x^6 + 2015x^5 + 2015x^3 + 2014x^2 + 2014x + 4056196$$

```
> factor(f);
```

$$(x^5 + x^2 + 2014)(x^6 + x^3 + x + 2014)$$

```
> a:=evalf[50](fsolve(f,x));
```

$$a := -4.5889717749000659646182620618546195972229082925560$$

```
> with(LinearAlgebra):
> M:=Matrix(6,shape=identity);
```

$$M := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> M1:=<M|<seq(round(evalf[50](10^20*a^k)),k=0..5)>>;
```

$$M1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 100000000000000000000 \\ 0 & 1 & 0 & 0 & 0 & 0 & -458897177490006596462 \\ 0 & 0 & 1 & 0 & 0 & 0 & 210586619508294616895 5 \\ 0 & 0 & 0 & 1 & 0 & 0 & -966376053095183604633 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 443467243159312509142 22 \\ 0 & 0 & 0 & 0 & 0 & 1 & -203505866195082946168955 \end{bmatrix}$$

```
> with(IntegerRelations):
> LLL(M1);
```

$$\begin{bmatrix} 2014 & 0 & 1 & 0 & 0 & 1 & 0 \\ -273 & 4668 & -2012 & 854 & -4263 & -1001 & 5965 \\ -605 & -2848 & -1717 & 4011 & 6370 & 1186 & 3899 \\ 563 & -5226 & -797 & 6052 & -2480 & -824 & 373 \\ -574 & -4342 & 4219 & -5509 & 1107 & 556 & 6911 \\ 746 & -3523 & -15296 & -3408 & 97 & 33 & 3421 \end{bmatrix}$$

Hátizsák probléma - Merkle-Hellman titkosítás

```
> S:=[7,11,19,39,79,159,329,649,1297,2663];add(k,k=S);
```

$$S := [7, 11, 19, 39, 79, 159, 329, 649, 1297, 2663]$$

$$5252$$

```
> p:=nextprime(5252);
```

$$p := 5261$$

```
> m:=3000;
```

$$m := 3000$$

```
> minv:=1/m mod p;
```

$$minv := 4663$$

```
> publikus:=[seq((S[k]*m) mod p,k=1..nops(S))];
```

$$publikus := [5217, 1434, 4390, 1258, 255, 3510, 3193, 430, 3121, 2802]$$

```
> uzenet:=[1,0,1,0,1,0,1,0,1,0];
```

$$uzenet := [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$$

```
> kodolt:=add(t,t=[seq(uzenet[k]*publikus[k],k=1..nops(S))]);
```

$$kodolt := 16176$$

```
> u1:=kodolt*minv mod p;
```

$$u1 := 1731$$

```
> hatizsak:=proc(H,s)
> local V,k,s1; V:=[seq(0,k=1..nops(H))];s1:=s;
> for k from nops(H) to 1 by -1 do
> if s1>=H[k] then V[k]:=1; s1:=s1-H[k]; end if;
> end do;
> V;
> end proc;
```

$$hatizsak := \mathbf{proc}(H, s)$$

$$\mathbf{local}\ V, k, s1;$$

$$V := [seq(0, k = 1 ..nops(H))];$$

$$s1 := s;$$

$$\mathbf{for}\ k\ \mathbf{from}\ nops(H)\ \mathbf{by}\ -1\ \mathbf{to}\ 1\ \mathbf{do}\ \mathbf{if}\ H[k] <= s1\ \mathbf{then}\ V[k] := 1;\ s1 := s1 - H[k]\ \mathbf{end\ if}\ \mathbf{end\ do};$$

$$V$$

$$\mathbf{end\ proc}$$

```
> hatizsak(S,u1);
```

$$[1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$$

```
> with(LinearAlgebra):
> M:=Matrix(10,shape=identity);
```

Hátizsák probléma - Merkle-Hellman titkosítás

$$M := \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

```
> M1:=convert(<M,Vector[row](10,fill=0)>,matrix);
```

$$M1 := \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

```
> M2:=<M1|Vector[column](11,[seq(publikus[k],k=1..10)],fill=-kodolt)>:
  convert(M2,matrix);
```

$$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&5217 \\ 0&1&0&0&0&0&0&0&0&0&1434 \\ 0&0&1&0&0&0&0&0&0&0&4390 \\ 0&0&0&1&0&0&0&0&0&0&1258 \\ 0&0&0&0&1&0&0&0&0&0&255 \\ 0&0&0&0&0&1&0&0&0&0&3510 \\ 0&0&0&0&0&0&1&0&0&0&3193 \\ 0&0&0&0&0&0&0&1&0&0&430 \\ 0&0&0&0&0&0&0&0&1&0&3121 \\ 0&0&0&0&0&0&0&0&0&1&2802 \\ 0&0&0&0&0&0&0&0&0&0&-16176 \end{bmatrix}$$

```
> with(IntegerRelations):
> convert(LLL(M2),matrix);
```

Hátizsák probléma - Merkle-Hellman titkosítás

$$\begin{bmatrix}
0 & -1 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & -1 \\
1 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 0 & -1 \\
1 & 0 & -1 & 0 & -2 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 & -2 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 2 & -1 & 0 & 1 & -1 & 0 & 1 & 0 & 1 \\
-1 & -1 & -1 & 1 & 0 & 2 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 1 & 2 & -1 \\
-1 & -1 & 1 & -1 & 1 & 0 & 2 & 0 & -1 & 0 & 1 \\
2 & -1 & -1 & 0 & 0 & -2 & -1 & 0 & 0 & 2 & 1
\end{bmatrix}$$

Index kalkulus

Additív és multiplikatív csoportok

$$(G, +) \Longrightarrow ng_1 = g_2 \quad \text{additív}$$

$$(H, \times) \Longrightarrow h_1^n = h_2 \quad \text{multiplikatív}$$

$\mathbb{F}_q$ multiplikatív csoportban:

$$a^k \equiv p_1^{t_1} p_2^{t_2} \cdots p_m^{t_m} \pmod{q}$$

$$k \equiv t_1 \log_a p_1 + t_2 \log_a t_2 + \ldots + t_m \log_a p_m \pmod{q-1}$$

Index kalkulus

```
> q:=101;
```

$$q := 101$$

```
> nops({seq(7^k mod 101,k=1..101)});
```

$$100$$

```
> seq([k,ifactor(7^k mod 101)],k=1..20);
```

$$[1, (7)], [2, (7)^2], [3, (2)^3 (5)], [4, (2) (3) (13)], [5, (41)], [6, (5) (17)], [7, (2) (3)^2 (5)], [8, (2)^3 (3)], [9, (67)], [10, (5) (13)], [11, (3) (17)], [12, (2) (3)^3], [13, (3) (5)^2], [14, (2)^2 (5)], [15, (3) (13)], [16, (71)], [17, (3) (31)], [18, (3)^2 (5)], [19, (2)^2 (3)], [20, (2)^2 (3) (7)]$$

```
> with(LinearAlgebra[Modular]):
> M:=Mod(100,Matrix([[0,0,0,1],[3,0,1,0],[0,1,2,0],[3,1,0,0]]),integer[]);
```

$$M := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 3 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 3 & 1 & 0 & 0 \end{bmatrix}$$

```
> MI:=Inverse(100,M);
```

$$MI := \begin{bmatrix} 0 & 78 & 11 & 89 \\ 0 & 66 & 67 & 34 \\ 0 & 67 & 67 & 33 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

```
> b:=Mod(100,Matrix(4,1,[1,3,13,8]),integer[]);
```

$$b := \begin{bmatrix} 1 \\ 3 \\ 13 \\ 8 \end{bmatrix}$$

```
> Multiply(100,MI,b);
```

$$\begin{bmatrix} 89 \\ 41 \\ 36 \\ 1 \end{bmatrix}$$

```
> seq([k,ifactor(7^k*25 mod 101)],k=1..20);
```

$$[1, (2) (37)], [2, (13)], [3, (7) (13)], [4, (31)], [5, (3) (5)], [6, (2)^2], [7, (2)^2 (7)], [8, (5) (19)], [9, (59)], [10, (3)^2], [11, (3)^2 (7)], [12, (37)], [13, (3) (19)], [14, (2)^5 (3)], [15, (2) (3) (11)], [16, (2) (29)], [17, (2)], [18, (2) (7)], [19, (2) (7)^2], [20, (2)^4 (5)]$$

```
> x:=(41+36-5) mod 100;
```

$$x := 72$$

```
> 7^72 mod 101;
```

$$25$$

```
> IndexCalc:=proc(a,b,q,FB) local B,M,M1,k,l,r,LOGS,logseq;
> with(numtheory): with(padic):
```

Index kalkulus

```
> with(LinearAlgebra[Modular]):
> B:=Mod(q-1,Matrix(nops(FB),1),integer[]);
> M:=Mod(q-1,Matrix(nops(FB),nops(FB)),integer[]);
> M1:=M; k:=1; l:=1; r:=[seq(Rank(p,M1),p=factorset(q-1))];
> while add(s,s=[seq(Rank(p,M),p=factorset(q-1))])<nops(factorset(q-1))*nops
  (FB) do
> if evalb(factorset(a^k mod q) subset FB) then M1[l]:=Mod(q-1,Matrix(1,nops
  (FB),[seq(ordp(a^k mod q,f),f=FB)]),integer[]); print(M1,r,[seq(Rank(p,
  M1),p=factorset(q-1))],l,k); end if;
> if add(s,s=[seq(Rank(p,M1),p=factorset(q-1))])-add(s,s=r)=nops(factorset
  (q-1)) then M:=M1; B[l]:=k; l:=l+1; r:=[seq(Rank(p,M1),p=factorset(q-1))]
  end if; k:=k+1;
> end do;
> M; print(FB); LOGS:=Multiply(q-1,Inverse(q-1,M),B); print(LOGS);
> k:=1; while not evalb(factorset((a^k*b) mod q) subset FB) do
> k:=k+1; print(k); print(factorset((a^k*b) mod q));
> end do;
> logseq:=[seq(ordp((a^k*b) mod q,f),f=FB)]; print(logseq);
> (add(k,k=[seq(LOGS[t,1]*logseq[t],t=1..nops(LOGS))])-k) mod (q-1);
> end proc;
```

```
IndexCalc := proc(a, b, q, FB)
    local B, M, M1, k, l, r, LOGS, logseq;
    with(numtheory);
    with(padic);
    with(LinearAlgebra[Modular]);
    B := `LinearAlgebra:-Modular`:-Mod(q - 1, Matrix(nops(FB), 1), integer[ ]);
    M := `LinearAlgebra:-Modular`:-Mod(q - 1, Matrix(nops(FB), nops(FB)), integer[ ]);
    M1 := M;
    k := 1;
    l := 1;
    r := [seq(`LinearAlgebra:-Modular`:-Rank(p, M1), p = numtheory:-factorset(q - 1))];
    while add(s, s = [seq(`LinearAlgebra:-Modular`:-Rank(p, M), p = numtheory:-factorset(q - 1))])
     < nops(numtheory:-factorset(q - 1)) * nops(FB) do
        if evalb(subset(numtheory:-factorset(mod(a^k, q)), FB)) then
            M1[l] := `LinearAlgebra:-Modular`:-Mod(q - 1, Matrix(1, nops(FB), [seq(padic:-ordp(mod(a
            ^k, q), f), f = FB)]), integer[ ]);
            print(M1, r, [seq(`LinearAlgebra:-Modular`:-Rank(p, M1), p = numtheory:-factorset(q - 1))],
            l, k)
        end if;
        if add(s, s = [seq(`LinearAlgebra:-Modular`:-Rank(p, M1), p = numtheory:-
        factorset(q - 1))]) - add(s, s = r) = nops(numtheory:-factorset(q - 1)) then
            M := M1;
            B[l] := k;
            l := l+1;
            r := [seq(`LinearAlgebra:-Modular`:-Rank(p, M1), p = numtheory:-factorset(q - 1))]
```

Index kalkulus

```
        end if;
        k := k+1
    end do;
    M;
    print(FB);
    LOGS := `LinearAlgebra:-Modular`:-Multiply(q-1, `LinearAlgebra:-Modular`:-Inverse(q-1, M), B);
    print(LOGS);
    k := 1;
    while not evalb(subset(numtheory:-factorset(mod(a^k*b, q)), FB)) do
        k := k+1; print(k); print(numtheory:-factorset(mod(a^k*b, q)))
    end do;
    logseq := [seq(padic:-ordp(mod(a^k*b, q), f), f=FB)];
    print(logseq);
    mod(add(k, k=[seq(LOGS[t, 1]*logseq[t], t=1..nops(LOGS))])-k, q-1)
end proc
> IndexCalc(7,24,101,{2,3,5,7,11});
```

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [0, 0], [1, 1], 1, 1$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [1, 1], [1, 1], 2, 2$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [1, 1], [2, 2], 2, 3$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [2, 2], [2, 3], 3, 7$$

Index kalkulus

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [2, 2], [3, 3], 3, 8$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [3, 3], [3, 4], 4, 12$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, [3, 3], [4, 4], 4, 13$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \end{bmatrix}, [4, 4], [4, 4], 5, 14$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \end{bmatrix}, [4, 4], [4, 4], 5, 18$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 \end{bmatrix}, [4, 4], [4, 4], 5, 19$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \end{bmatrix}, [4, 4], [4, 4], 5, 20$$

Index kalkulus

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \end{bmatrix}, [4, 4], [4, 4], 5, 23$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{bmatrix}, [4, 4], [5, 5], 5, 24$$

$$\{2, 3, 5, 7, 11\}$$

$$\begin{bmatrix} 89 \\ 41 \\ 36 \\ 1 \\ 57 \end{bmatrix}$$

$$2$$

$$\{5, 13\}$$

$$3$$

$$\{3, 17\}$$

$$4$$

$$\{2, 3\}$$

$$[1, 3, 0, 0, 0]$$

$$8$$

Elliptikus görbék - összeadás

Elliptikus görbe

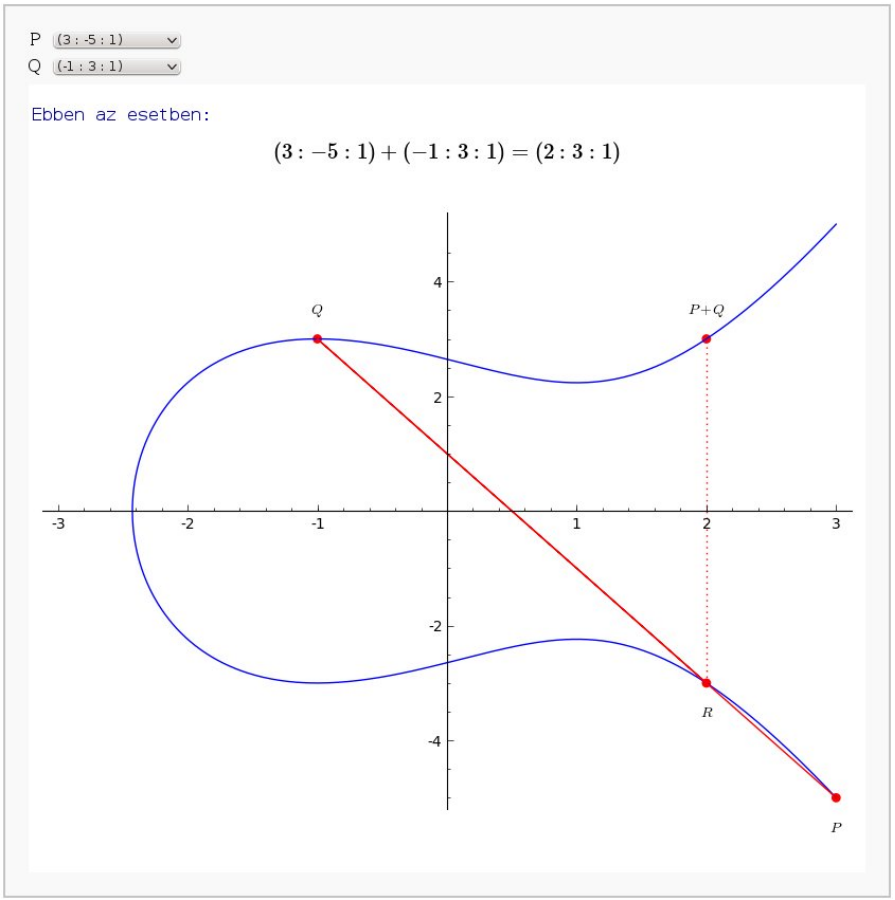
$$E: \quad y^2 = x^3 + ax + b$$

$$E(\mathbb{Q}) = \{(x, y) \in \mathbb{Q}^2 : y^2 = x^3 + ax + b\} \cup \{\infty\}.$$

A görbe pontjain bevezetünk egy összeadás műveletet.

Elliptikus görbék - összeadás

$$y^2 = x^3 - 3x + 7$$

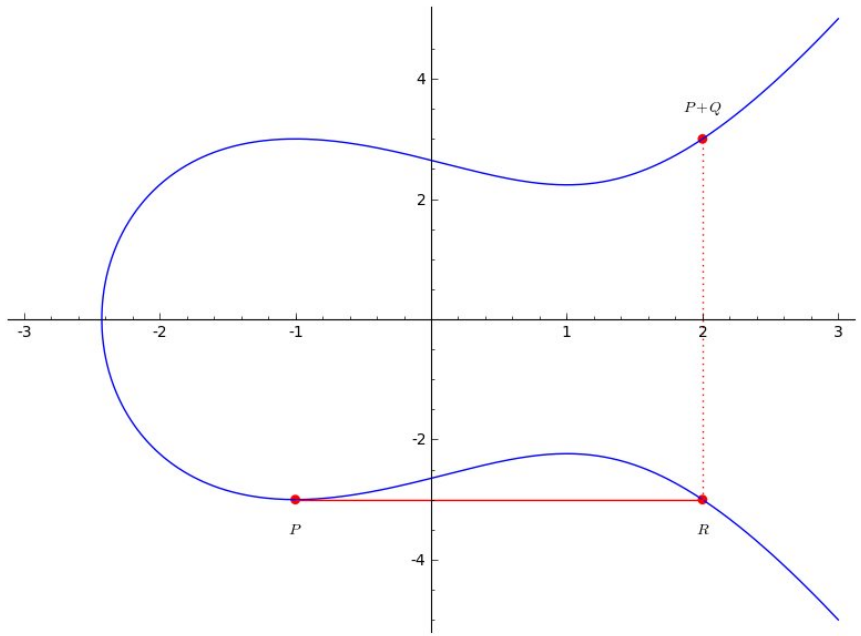


Elliptikus görbék - összeadás

P (-1 : -3 : 1)
Q (-1 : -3 : 1)

Ebben az esetben:

$$(-1:-3:1)+(-1:-3:1)=(2:3:1)$$

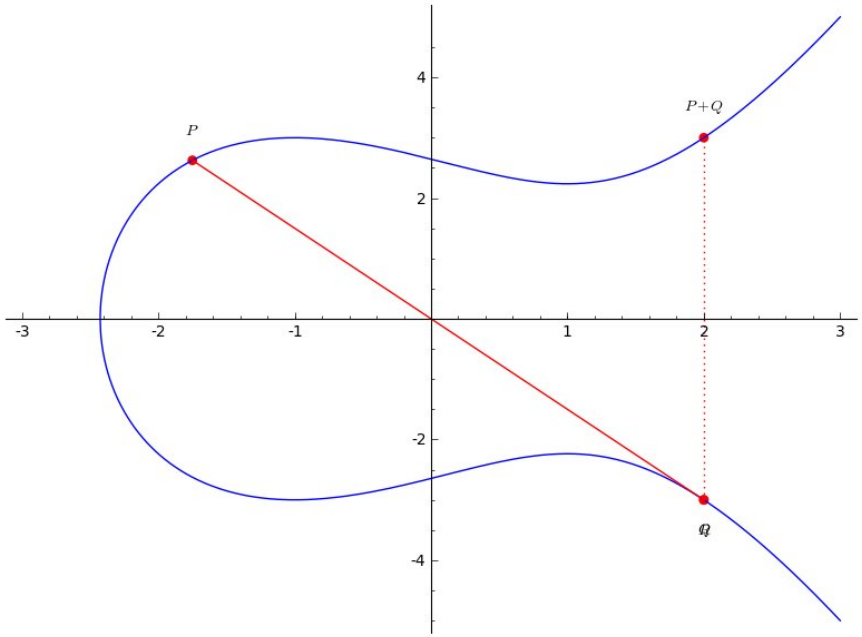


Elliptikus görbék - összeadás

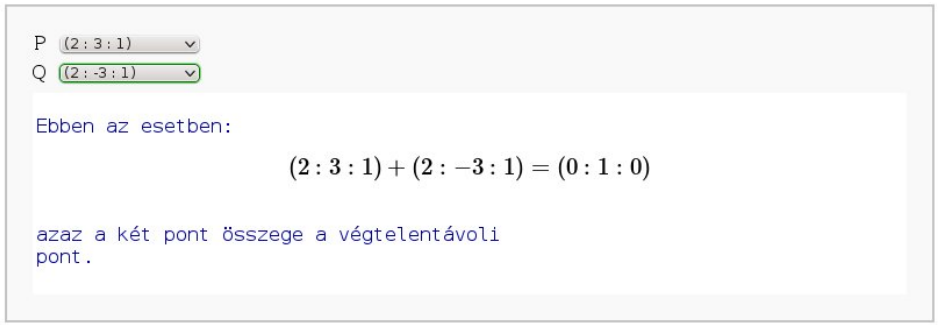
P (-7/4 : 21/8 : 1)
Q (2 : -3 : 1)

Ebben az esetben:

$$\left(-\frac{7}{4} : \frac{21}{8} : 1\right) + (2 : -3 : 1) = (2 : 3 : 1)$$



Elliptikus görbék - összeadás



Most bemutatjuk hogyan lehet a Maple segítségével az összeadást elvégezni adott elliptikus görbék esetében. A módszer működik $\mathbb{Z}_N$ felett is, de összetett N esetében előfordulhat, hogy nem tudunk kiszámítani bizonyos nevezőket. Ezt fel tudjuk használni faktorizációra.

Elliptikus görbék - összeadás

```
> elliptic_add := proc(p1,p2,E) local g,inv,x1,x2,x3,y1,y2,y3,a,b,n;
    x1 := p1[1]; x2 := p2[1]; y1 := p1[2]; y2 := p2[2];
    a := E[1]; b := E[2]; n := E[3];

    if (x1=infinity and y1=infinity) then [x2,y2];
     elif (x2=infinity and y2=infinity) then [x1,y1];
     elif (x1=x2 and y1 <> y2) then [infinity,infinity];

     elif (x1=x2 and y1=y2 and y1<>0) then
       g := igcdex(2*y1,n,inv);
       if g = 1 then
         x3 := ((3*x1^2+a)*inv)^2-2*x1  mod n;
         y3 := ((3*x1^2+a)*inv)*(x1-x3)-y1  mod n;
         [x3,y3];
       else g; end if;
     elif (x1=x2 and y1=y2 and y1=0) then
       [infinity,infinity];

     else
       g := igcdex(x2-x1,n,inv);
       if g = 1 then
         x3 := ((y2-y1)*inv)^2-x1-x2 mod n;
         y3 := (y2-y1)*inv*(x1-x3)-y1 mod n;
         [x3,y3];
       else g; end if;
   end if;
   end proc;
```

```
elliptic_add := proc(p1, p2, E)
    local g, inv, x1, x2, x3, y1, y2, y3, a, b, n;
    x1 := p1[1];
    x2 := p2[1];
    y1 := p1[2];
    y2 := p2[2];
    a := E[1];
    b := E[2];
    n := E[3];
    if x1 = infinity and y1 = infinity then
        [x2, y2]
    elif x2 = infinity and y2 = infinity then
        [x1, y1]
    elif x1 = x2 and y1 <> y2 then
        [infinity, infinity]
    elif x1 = x2 and y1 = y2 and y1 <> 0 then
        g := igcdex(2 * y1, n, inv);
        if g = 1 then
            x3 := mod((3 * x1^2 + a)^2 * inv^2 - 2 * x1, n); y3 := mod((3 * x1^2 + a) * inv
            * (x1 - x3) - y1, n);
            [x3, y3]
        else
            g
        end if
    elif x1 = x2 and y1 = y2 and y1 = 0 then
        [infinity, infinity]
    else
        g := igcdex(x2 - x1, n, inv);
```

Elliptikus görbék - pont többszöröse

```
        if g = 1 then
            x3 := mod((y2-y1)^2*inv^2-x1-x2, n); y3 := mod((y2-y1)*inv
             *(x1-x3)-y1, n);
            [x3, y3]
        else
            g
        end if
    end if
end proc
> elliptic_mul := proc(p1,k,E) local pn,ps,d,r;
    pn := p1; r := irem(k,2);
    if r = 1 then ps := p1; else ps := [infinity,infinity]; end if;
    d := (k-r)/2;

    while (d > 0 and (not (type(ps,integer) or type(pn,integer)) and (pn <>
  [infinity,infinity]))) do
     r := irem(d,2);
     pn := elliptic_add(pn,pn,E);
     if r = 1 then ps := elliptic_add(ps,pn,E); end if;

     d := (d-r)/2;
    end do;

    if type(pn,integer) then pn;
     else ps; end if;
   end proc;
elliptic_mul := proc(p1, k, E)
    local pn, ps, d, r;
    pn := p1;
    r := irem(k, 2);
    if r = 1 then ps := p1 else ps := [infinity, infinity] end if;
    d := 1/2*k-1/2*r;
    while 0 < d and not (type(ps, integer) or type(pn, integer)) and pn<>[infinity, infinity] do
        r := irem(d, 2);
        pn := elliptic_add(pn, pn, E);
        if r = 1 then ps := elliptic_add(ps, pn, E) end if;
        d := 1/2*d-1/2*r
    end do;
    if type(pn, integer) then pn else ps end if
end proc
> N:=65^64+64;
N :=
    10628080253478339771393136453028132434200582652706852594290949688815364814007541349 7\
    38517055811826139688491821289068 9
> elliptic_mul([0,1],2*3*5*7*11*13*17*19,[21,1,N]);
                                    12653
> N mod 12653;
                                      0
> N/12653;
8399652456712510686314025490419767987197172727975067252265035713913984678738276574518704\
    6991320842011289748054 13
> k:=1; while nops(elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43,
  [k,1,N/12653]))<>1 do k:=k+1; end do:
                                    k := 1
> k;
```

Elliptikus görbék - faktorizáció

```
                                   31
> elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43,[31,1,N/12653]);
                                  136573
> k:=1; while nops(elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43
  [k,1,N/(12653*136573)]))<>1 do k:=k+1; end do:
                                   k := 1
> k;
                                   131
> elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43,[131,1,N/(12653*
  136573)]);
                                  89009
> N/(12653*136573*89009);
690975336278659420330330852565411677475191672511680407054887907340368896259444326042140\5
  05656609610009
> k:=1; while nops(elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43
  [k,1,N/(12653*136573*89009)]))<>1 do k:=k+1; end do:
                                   k := 1
> k;
                                   161
> elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43,[161,1,N/(12653*
  136573*89009)]);
                                  929501
> N/(12653*136573*89009*929501);
74338310155519942456256728348373124663146319639428080986990644156420369236767289765383\84
  63494509
> k:=1; while nops(elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43
  [k,1,N/(12653*136573*89009*929501)]))<>1 do k:=k+1; end do:
                                   k := 1
> k;
                                   376
> elliptic_mul([0,1],2*3*5*7*11*13*17*19*23*29*31*37*41*43,[376,1,N/(12653*
  136573*89009*929501)]);
                                 2500297
> N/(12653*136573*89009*929501*2500297);
29731791925327248105427766520686592298093514346266895887564814962550596683820877985848\81
  97
```