

Rezolvarea ecuațiilor diofantice

Orice congruență $ax_1 + c \equiv 0 \pmod{b}$ se poate scrie ca o ecuație $ax_1 + bx_2 + c = 0$ (în care $a \neq 0$), $b \geq 1$ și c, x_1, x_2 sunt numere întregi. Dacă a, b, c sunt numere întregi date și x_1 și x_2 sunt considerate necunoscute, problema se reduce la găsirea soluțiilor întregi ale unei ecuații liniare cu coeficienți întregi. Dacă $f(x_1, \dots, x_n)$ este un polinom în x_1, \dots, x_n cu coeficienți întregi, atunci ecuația $f(x_1, \dots, x_n) = A$ se numește **diofantică** dacă soluțiile ei sunt numere întregi. Denumirea acestor ecuații derivă de la numele matematicianului grec **Diofantos** din Alexandria. Dacă o astfel de ecuație admite soluții, atunci ea admite o infinitate de n -upluri care o satisfac.

În continuare se va trata cazul $n=2$: $ax+by=c$

Dacă a și b sunt numere prime între ele și x_0, y_0 constituie o soluție pentru $ax+by=c$, atunci totalitatea soluțiilor se poate reprezenta sub forma: $x = x_0 + bt, y = y_0 - at$, unde t este un număr întreg oarecare. O soluție a ecuației se poate obține cu ajutorul penultimei fracții de aproximare pentru reprezentarea sub formă de fracție continuă a lui a/b . Considerând că penultima fracție este m/n , $x_0 = nc, y_0 = -mc$.

Exemplu:

Fie ecuația: $43x + 19y = 2$.

Fracțiile de aproximare ale lui $43/19$ sunt: $7/3, 9/4, 43/19$.

Din fracția $9/4$ se obține $x_0 = 4 \cdot 2 = 8, y_0 = -9 \cdot 2 = -18$. Astfel, soluția generală se poate scrie de forma: $x = 8 + 19t$ și $y = -18 - 43t$, unde t este un număr oarecare.

Implementare

Algoritmul de mai sus este valabil, după cum am precizat, în cazul când cele 2 numere a și b sunt prime între ele. Dacă dorim rezolvarea unei ecuații în care cele 2 numere nu sunt neapărat prime între ele, se poate proceda în felul următor: se calculează cel mai mare divizor comun al lor (sigur este diferit

de 1), iar apoi se evaluează dacă ecuație poate sau nu avea soluții, în funcție de valoarea lui c . Dacă c este divizibil cu cmmdc-ul celor 2 numere, atunci se simplifică întreaga ecuație cu cmmdc și problema se reduce la cea prezentată mai sus. Dacă c nu se împarte exact la cmmdc, atunci putem spune că ecuația nu are soluții întregi.

Pe lângă aceasta, intervin o serie de cazuri critice în care algoritmul de mai sus nu poate fi aplicat, cum ar fi de exemplu cazurile în care nu există penultima fracție. Dar se poate calcula soluția și în aceste cazuri:

- $a=0, b=0$

În acest caz soluția depinde valoarea lui c :

- $c=0 \Rightarrow x$ și y poate fi orice număr întreg
- $c \neq 0 \Rightarrow$ nu există soluții

- $a=0, b \neq 0$

Ecuația devine: $by=c$. Deci y se poate calcula, ținând însă cont că vorbim numai de numere întregi.

- $a \neq 0, b=0$

Analog cu cazul anterior.

Dacă unul dintre numerele a sau b are valoarea 1 nu se mai poate vorbi de penultima fracție de aproximare, deci și aceste cazuri trebuie tratate separat.

O altă observație este aceea că fracția de aproximare (m/n) aproximează fracția (a/b) în plus sau în minus. De aceea în la sfârșit trebuie să corectez rezultatul în funcție de aceasta, ținând seama și de semnul fracției a/b .

Sursa programului

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

long int v[100];

//obține penultima fracție de aproximare
void get_mn(long int &a,long int &b,int k)
{
```

```

    if (k>0) { long int aux=v[k-1]*b+a;
                a=b;b=aux;
                get_mn(a,b,k-1);
            }
        else a=a+b-(b=a);
    }

long int _cmmdc(long int a,long int b)
{
    while (a!=b)
        if (a>b) a-=b;
            else b-=a;
    return a;
}

void stop()
{
    printf("Nu exista solutii...\n");
}

int solutii(long int a,long int b,long int c,
            long int &x0,long int &n1,long int
&y0,long int &n2)
{
    long int m,n,cmmdc=1,_a,_b;
    int nr=-1;

    _a=labs(a);_b=labs(b);
    if ((_a>1)&&(_b>1)) cmmdc=_cmmdc(_a,_b);
    if (cmmdc!=1){
        if (c%cmmdc) {stop();return 0;}
        else a/=cmmdc,b/=cmmdc,c/=cmmdc;
    }
    m=a;n=b;
    if (!(a*b))
        if (!a) if (!b)
            {
                //0=c
                if (!c) printf("x,y∈Z\n");
                else stop();
                return 0;
            }
        else

```

```

    { // by=c
      if (c%b) stop();
        else { printf("x∈Z\n");
              printf("y=%ld\n",c/b);
            }
      return 0;
    }
  else
  { //ax=c
    if (c%a) stop();
      else {printf("x=%ld\n",c/a);
            printf("y∈Z\n");
          }
    return 0;
  }
}

if (labs(m)==1)
  { // inseamna ca este de forma ±x+b*y=c, deci nu
  exista
    // penultima fractie de aproximare
    x0=c*m;n1=-b*m;
    y0=0;n2=1;
    return 1;
  }
else
  if (labs(n)==1)
    { // inseamna ca este de forma a*x±y=c, deci nu
  exista
    // penultima fractie de aproximare
    x0=0;n1=1;
    y0=c*n;n2=-a*n;
    return 1;
  }

// m si n sunt diferite de 1, si diferite intre
ele
m=labs(m);n=labs(n);
while (m!=1){
  if (m>n){ v[++nr]=m/n;
            m-=m/n*n;
          }
  else if (m<n) m=m+n-(n=m);
}

```

```

}

n=v[nr];
get_mn(m,n,nr);
if (_a<_b) m=m+n-(n=m);

//in functie de fractia de aproximare corectez
rezultatul
if (a*b>0) if (_a*n<_b*m) c=-c;
if (a*b<0) if (_a*n>_b*m) c=-c;
if (a<0) m=-m;
if (b<0) n=-n;

x0=n*c;n1=b;
y0=-m*c;n2=-a;
return 1;
}

void main(void)
{

long int a,b,c,x0,y0,n1,n2;

do{
clrscr();
puts("Se rezolva ecuatia: ax+by=c, a,b,x,y,cÎZ");
printf("a=");scanf("%ld",&a);fflush(stdin);
printf("b=");scanf("%ld",&b);fflush(stdin);
printf("c=");scanf("%ld",&c);fflush(stdin);
printf("-----
\n");

if (solutii(a,b,c,x0,n1,y0,n2)){
if (!x0) if (labs(n1)!=1) printf("x=%ldt\n",n1);
else printf("x=%st\n",n1<0?"-
":"");
else if (labs(n1)!=1)
printf("x=%ld%+ldt\n",x0,n1);
else
printf("x=%ld%st\n",x0,n1<0?"-":"");

if (!y0) if (labs(n2)!=1) printf("y=%ldt\n",n2);

```

```
                                else printf("y=%st\n",n2<0?"-
":"");
    else if (labs(n2)!=1)
printf("y=%ld%+ldt\n",y0,n2);
                                else
printf("y=%ld%st\n",y0,n2<0?"-":"");

    printf("unde t este un numar intreg\n");}
    puts("Apasati o tasta (x pentru a termina)...");
    c=getch();
    }while (c!='x');
}
```