1. Teoretické poznatky

Rabin-Millerov test je v praxi najviac používaný test prvočíselnosti. Test je založený na nasledujúcom fakte.

Veta:

Nech je *n* náhodné prvočíslo a nech $n - 1 = 2^s r$ kde *r* je náhodné. Nech *a* je náhodné číslo pre ktoré platí: *gcd(a,n)=1*. Potom tiež

$$a^r \equiv 1 (\mathrm{mod}\, n)$$

alebo

$$a^{2^j r} \equiv -1 (\mathrm{mod}\, n)$$

pre každé *j*, pričom $0 \leq j \leq s - 1$.
Na tomto fakte je založená nasledujúca definícia:

Definícia:

Nech *n* je náhodné združené číslo a nech $n - 1 = 2^s r$ kde *r* je náhodné. Nech *a* je číslo z intervalu [1,n – 1].

1.) Ak

$$a^r \neq 1 \ (\mathrm{mod}\ n)$$

alebo

$$a^{2^j r} \neq -1 (\mathrm{mod}\, n)$$

pre každé j, pričom $0 \leq j \leq s - 1$, potom *a* je „silným dôkazom“ kompozitnosti *n*.

2.) V opačnom prípade, t.j. ak

$$a^r \equiv 1 (\mathrm{mod}\, n)$$

a ak

$$a^{2^j r} \equiv -1 (\mathrm{mod}\, n)$$

pre j, $0 \leq j \leq s - 1$, potom *n* je „silné pseudoprvočíslo na báze a“.

Príklad:

Uvažujme zložené číslo n=91 (=7×13). Pretože 91-1=90=2×45, s=1 a r=45. Potom $9^r = 9^{45} \equiv 1 (\mathrm{mod}\ 91)$, 91 je „silné pseudoprvočíslo na báze 9“. Zoznam všetkých *a,* ktoré pre zložené číslo dávajú výsledok testu, že je to prvočíslo:

{1,9,10,12,16,17,22,29,38,53,62,69,74,75,79,81,82,90}.

Veta:

Ak *n* je náhodné zložené číslo, potom asi 1/4 všetkých čísiel a, $1 \leq a \leq n - 1$, sú bázami poskytujúcimi nepravdivý dôkaz kompozitnosti *n*. Ak n≠9, počet báz poskytujúcich nepravdivý dôkaz kompozitnosti n je Φ(n)/4, kde Φ je Eulerova phi funkcia.

2. Algoritmus Rabin-Millerovho testu

VSTUP: náhodné číslo n ≥ 3 a parameter t ≥ 1.
VÝSTUP: odpoveď „prvočíslo“ alebo „zložené“ na otázku: „Je *n* prvočíslo?“

1. Napíš: $n-1=2^s r$ také, že *r* je náhodné.
2. Pre *i* od 1 do t vykonaj:
   2.1 Zvoľ náhodné číslo a, $2 \le a \le n-2$.
   2.2 Počítaj $y = a^r \bmod n$ pomocou algoritmu 2.143.
   2.3 Ak $y \neq 1$ a $y \neq n-1$ potom nastav $j \leftarrow 1$.
   Pokiaľ $j \le s-1$ a $y \neq n-1$ vykonaj nasledovné: počítaj $y \leftarrow y^2 \bmod n$.
   Ak y = 1, potom vráť (zložené).
   $j \leftarrow j+1$.
   Ak $y \neq n-1$, potom vráť (zložené).
3. Vráť („prvočíslo“).

Predošlý algoritmus testuje či každá báza *a* splňuje podmienky definície.
V kroku 2.3, ak y=1, potom $a^{2^j r} \equiv 1 (\bmod n)$. Pretože toto je ten istý prípad ako

$$a^{2^{j-1} r} \neq \pm 1 (\bmod n)$$

to vyplýva z vety, že *n* je zložené (v skutočnosti $\gcd(a^{2^{j-1} r} - 1, n)$ je netriviálny faktor n).
V 2.3 kroku, ak $y \neq n-1$, tak *a* je silný dôkaz pre kompozitnosť *n*.

Veta:

Pre ľubovoľné náhodné zložené číslo *n* je pravdepodobnosť, že Rabin-Millerov test *n* označí ako prvočíslo je menšia ako $(1/4)^t$.

Poznámka:

Pre väčšinu zložených čísiel *n*, počet báz poskytujúcich nepravdivý dôkaz kompozitnosti n je oveľa menší ako horná hranica z Φ(n)/4. Preto pravdepodobnosť, že Rabin-Millerov test označí n nesprávne je oveľa menšia ako $(1/4)^t$ pre väčšinu kladných čísiel *n*.

Príklad:

Jedinými bázami poskytujúcimi nepravdivý dôkaz kompozitnosti pre zložené číslo n=105 (=3×5×7) sú 1 a 104. Všeobecne ak k ≥ 2 a *n* je produkt prvých k rôznych prvočísiel, potom sú len 2 „bázy poskytujúce nepravdivý dôkaz kompozitnosti“, a to 1 a n – 1.

Poznámka:

Ak a_1 a a_2 sú bázy poskytujúce nepravdivý dôkaz kompozitnosti *n*, potom ich súčin $a_1 a_2$ je veľmi pravdepodobne ale nie s istotou tiež bázou poskytujúcou nepravdivý dôkaz kompozitnosti *n*.

Definícia:

Nech $p_1, p_2, \ldots, p_t$ sú prvočísla. Potom ψ_t je definované ako najmenšie kladné zložené číslo, ktoré je silné pseudoprvočíslo pre všetky bázy $p_1, p_2, \ldots, p_t$.
Počet ψ_t môže byť vysvetlený nasledovne: aby bola určená prvočíselnosť ľubovoľného čísla $n < \psi_t$ je dostatočné aplikovať Rabin-Millerov algoritmus na *n* s bázami *a* existujúcimi pre prvých t prvočísel. S použitím týchto báz, výsledok Rabin-Millerovho testu bude vždy pravdivý. Nasledujúca tabuľka udáva hodnoty ψ_t pre $1 \le t \le 8$.

Tabuľka:

t	φ_t
1	2047
2	1373653
3	25326001
4	3215031751
5	2152302898747
6	3474749660383
7	341550071728321
8	341550071728321

Vzorový príklad pre prvočíslo:

Zvolíme si jedno prvočíslo, nech n = 7.
Musí platiť podmienka: $n - 1 = 2^s.r$, pričom sa treba snažiť 2^s maximalizovať; dosiahnuť aby bolo čo možno najväčšie. Po dosadení konkrétnych hodnôt: $7 - 1 = 2^1.3$
Pre i = 1 zvolíme *a* z intervalu $2 < a < n - 2$; konkrétne a = 3.
Vypočítame $y = a^r \bmod n$; $y = 3^3 \bmod 7 = 6 \bmod 7 = 6$
Porovnávame: $y \neq 1$ a $y \neq n - 1$, avšak y = 6 takže nastavíme j = 1.
Pre i = 2 zvolíme *a* z intervalu $2 < a < n - 2$; konkrétne a = 4.
Vypočítame $y = a^r \bmod n$; $y = 4^3 \bmod 7 = 1$
Porovnávame: $y \neq 1$ a $y \neq n - 1$, avšak y = 1 takže nastavíme j = 1.
Pre túto hodnotu a - čka je vylúčená možnosť, že *n* je zložené číslo. Zvýšime hodnotu *i*.
Pre i = 3 si zvolíme a = 5 ale táto voľba už nevyhovuje podmienke $2 < a < n - 2$ takže končíme cyklus. Výsledok je, že *n* je prvočíslo.

Vzorový príklad pre zložené číslo:

Zvolíme si jedno zložené číslo, nech n = 9.
Musí platiť podmienka: $n - 1 = 2^s.r$, pričom treba sa snažiť 2^s maximalizovať; dosiahnuť aby bolo čo možno najväčšie. Po dosadení konkrétnych hodnôt: $9 - 1 = 2^3.1$
Pre i = 1 zvolíme *a* z intervalu $2 < a < n - 2$; konkrétne a = 3.
Vypočítame $y = a^r \bmod n$; $y = 3^1 \bmod 9 = 3$
Porovnávame: $y \neq 1$ a $y \neq n - 1$, keďže y = 3, nastavíme j = 1.
Overujeme podmienky: $j \leq s - 1$, $j \leq 2$; $y \neq n - 1$, $y \neq 8$. Keďže platia, vypočítame $y = y^2 \bmod n$.
Konkrétne $y = 3^2 \bmod 9 = 0$. Testujeme, či y = 1. Keďže $y \neq 1$, nepotvrdilo sa, že je to zložené číslo.
Zvýšime j o 1 t.j. $j \leftarrow j + 1$; j = 2.
Overujeme podmienky: $j \leq s - 1$, $j \leq 2$; $y \neq n - 1$, $y \neq 8$. Keďže platia, vypočítame $y = y^2 \bmod n$.
Konkrétne $y = 0^2 \bmod 9 = 0$. Testujeme, či y = 1. Keďže $y \neq 1$, nepotvrdilo sa, že je to zložené číslo.
Zvýšime j o 1 t.j. $j \leftarrow j + 1$; j = 3.
Teraz už nie je splnená podmienka: $j \leq s - 1$, skončíme aktuálny cyklus.
Otestujeme, či $y \neq n - 1$, $y \neq 8$ => číslo je zložené. Existujú také čísla a, ktoré vyhovujú podmienke pre určenie prvočíselnosti ale sú to zložené čísla. Preto musíme vykonať ten cyklus niekoľkokrát.

3. Zdrojový text algoritmu

```
//---------------------------------------------------------------------------
#include <vcl\vcl.h>
#pragma hdrstop
#include <math.h>
#include "Unit2.h"
//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TForm1 *Form1;

int modulo(int a, int x,int n);

//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
 int a=1,n,s=0,r,j=1,i;
 double p=1.0,cit,men,k,y;

 n=Form1->Edit1->Text.ToInt();

 r=1;

 while(s!=p)
 {
  k=(double(n)-1)/double(r);
  cit=log10(k);
  men=log10(2.0);
  p=cit/men;
  s=floor(p);
  r=r+2;
 }

  r=r-2;

 for(i=1;i<=n-3;i++)
 {
  a++;
  y=modulo(a,r,n);

  if(y!=1 && y!=n-1)
  {
   j=1;
```

```
  while(j<=s-1 && y!=n-1)
  {
   y=modulo(y,2,n);

   if(y==1)
   {
    Form1->Label3->Caption="Je to zložené číslo";
    return;
   }
   else
   {
    j++;
   }
  }

  if(y!=n-1)
  {
   Form1->Label3->Caption="Je to zložené číslo";
   return;
  }

 }
 Form1->ProgressBar1->Position=100*a/n;
}
Form1->ProgressBar1->Position=0;
Form1->Label3->Caption="Je to prvočíslo";

}
//---------------------------------------------------------------------------

// a mod(n)

int modulo(int a,int x,int n)
{
 unsigned __int64  m=1;
 int i,j,x2,v,p;
 int b[20],c[20],d[20],e[20],f[20],g[20];

 for (i=0;i<20;i++)
 {
  b[i]=c[i]=d[i]=e[i]=f[i]=g[i]=0;
 }

 x2=x; p=a; i=0;

 do
 {
  v=p%n;
  if(x2&1 == 1)
```

```
 {
  //m=m*v;
  b[i]=v;
  i++;
 }

 x2=x2>>1;

 p=pow(v,2);
}
while(x2!=0);

//m=m%n;

i=j=0;

do
{
 if(b[i+1]!=0)
  c[j]=(b[i]*b[i+1])%n;
 else
  c[j]=b[i];

 i=i+2;
 j++;
}
while(b[i]!=0);

i=j=0;

do
{
 if(c[i+1]!=0)
  d[j]=(c[i]*c[i+1])%n;
 else
  d[j]=c[i];

 i=i+2;
 j++;
}
while(c[i]!=0);

i=j=0;

do
{
 if(d[i+1]!=0)
  e[j]=(d[i]*d[i+1])%n;
 else
  e[j]=d[i];
```

```
 i=i+2;
 j++;
 }
 while(d[i]!=0);

 i=j=0;

 do
 {
 if(e[i+1]!=0)
  f[j]=(e[i]*e[i+1])%n;
 else
  f[j]=e[i];

 i=i+2;
 j++;
 }
 while(e[i]!=0);

 i=j=0;

 do
 {
 if(f[i+1]!=0)
  g[j]=(f[i]*f[i+1])%n;
 else
  g[j]=f[i];

 i=i+2;
 j++;
 }
 while(f[i]!=0);


 m=g[0]%n;

 return(m);
}
```