Criptografie - Tema 2

Popescu Paul - Constantin
Facultatea de Matematică

1. Găsiți numărul minim și maxim de pași pentru algoritmul lui Euclid. Explicați proprietățile.

Cel mai favorabil caz pentru algoritmul lui Euclid este atunci când unul dintre numere este un multiplu al celuilalt. De exemplu:

$$\text{cmmdc}(26, 13)$$

in care algoritmul se opreste dupa un singur pas, returnand 13 ($26 \bmod 13 = 0$).

Cel mai nefavorabil caz pentru algoritmul lui Euclid este atunci când cele 2 numere fac parte din sirul lui Fibonacci, deoarece sunt nevoie de $n-1$ pasi pentru a ajunge la un rezultat. Exemplu:

$$\text{cmmdc}(34, 21) =$$

$$\begin{aligned}
34 &= 1 \cdot 21 + 13 \\
21 &= 1 \cdot 13 + 8 \\
13 &= 1 \cdot 8 + 5 \\
8 &= 1 \cdot 5 + 3 \\
5 &= 1 \cdot 3 + 2 \\
3 &= 1 \cdot 2 + 1 \\
2 &= 1 \cdot 1 + 1
\end{aligned}$$

Proprietăți:

1. Eficienta: algoritmul are o complexitate logaritmica: $O(\log n)$, acest fiind foarte eficient.
2. Generalitate: funcționează pentru orice pereche de numere întregi pozitive.

2. Găsiți numărul de operații elementare pentru algoritmul lui Euclid.

Avand in vedere **Exercitiul 1**, algoritmul lui Euclid are o complexitate logaritmica. Deoarece fiecare pas reduce numărul mai mare la un rest mai mic decât cel anterior, rezulta ca numărul de pași este proporțional cu numărul mai mic dintre cele două. Deci putem spune ca complexitatea algoritmului lui Euclid este: $O\left(\log(\min(a, b))\right)$.

Pentru cazul cel mai nefavorabil (numerele fac parte din Sirul lui Fibonacci), descoperim o relatie interesanta cu numarul de aur $\phi \approx 1.618$ astfel:

Pentru exemplul de la **Exercitiul 1**, numarul de pasi este $\log_\phi(21) \approx 7$, chiar numarul de pasi al algoritmului.

3. Găsiți numărul de operații elementare pentru algoritmul lui Euclid extins.
Pentru varianta extinsa a algoritmului lui Euclid trebuie sa calculam suplimentar inca 2 recurente, aceastea fiind:

$$x_i = x_{i-2} - q_i x_{i+1} \quad \text{si} \quad y_i = y_{i-2} - q_i y_{i+1}$$

Deoarece aceste operatii sunt doar niste adunari si inmultiri, iar algoritmul parcurge acelasi numar de pasi ca in versiunea standard, complexitatea ramane aceiasi: $O(\log n)$ (n reprezentand $\min(a, b)$, valorile initiale).

4. Demonstrati formula $\sum_{d|n} \varphi(n) = n$.

5. Scrieți o variantă de cod pentru calculul funcției indicatoare a lui Euler. Asigurați-vă că funcționează și pentru numere mari. (nu apelați la librării)

`functieIndicatoareEuler.hpp`:

```cpp
#include <iostream>
using namespace std;

long long functieIndicatoareEuler(long long numar) {
   if (numar == 1) return 1;

   long long rezultat = numar;
   long long p = 2;

   while (p * p <= numar) {
      if (numar % p == 0) {
         while (numar % p == 0)
         numar /= p;
         rezultat -= rezultat / p;
      }
      if (p == 2)  p++;
      else p += 2;
   }

   if (numar > 1)
   rezultat -= rezultat / numar;

   return rezultat;
}
```

`main.cpp`:

```cpp
#include "functieIndicatoareEuler.hpp";

int main() {
   long long numar;
   cout << "Introdu un numar: ";
   cin >> numar;

   cout << "phi(" << numar << ") = " << functieIndicatoareEuler(numar) << endl;
   return 0;
}
```