Universidad Católica San Pablo

Facultad de Ingenierías y Arquitectura

TRABAJO DE INVESTIGACION E IMPLEMENTACION DEL ALGORITMO DE EUCLIDES Y ALGORITMO EXTENDIDO DE EUCLIDES

Alumnos:

Eduardo Arturo Espinoza Menacho

Anthony Erick Raúl Gamarra González

Semestre 2021 – I

Contenido

Introducción. -

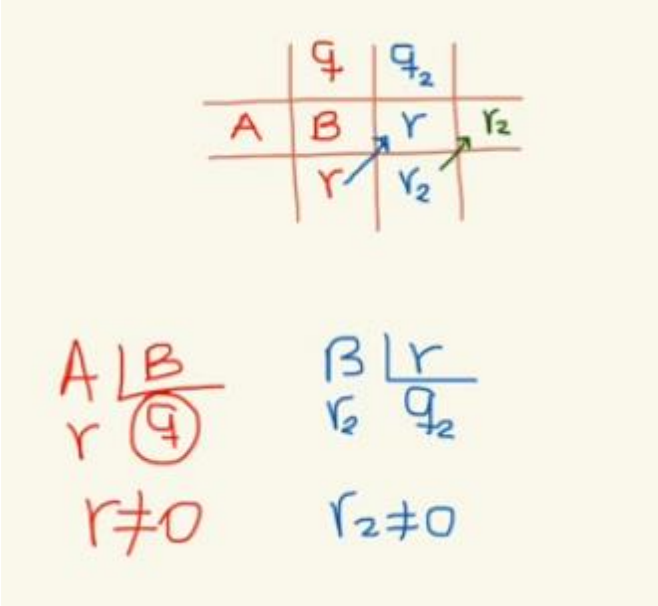
Hemos analizado 2 algoritmos que permiten el cálculo del máximo común divisor, tanto el algoritmo de Euclides, como el algoritmo extendido de Euclides. Este algoritmo tiene aplicaciones en diversas áreas como álgebra, teoría de números y ciencias de la computación, entre otras. Con unas ligeras modificaciones suele ser utilizado en computadoras electrónicas debido a su gran eficiencia. El objetivo de esta investigación es analizar los algoritmos y determinar, cual es más eficiente y rápido a la hora del trabajo.

I. Contenido Teórico

1.1. Algoritmo de Euclides:

Para ejemplificar la utilización de el algoritmo de Euclides, vamos a demostrar su forma de practica tanto matemáticamente, representada, escrita, tanto normal, como en pseudo código y en programación C++.

Lo primero para hacer el algoritmo de Euclides, hay que crear un esquema que separe los elementos de la división. Para de ahí operar y ordenar en el esquema. Según como se muestra la imagen. Para cuando el residuo final llegue a 0, Hallar el MCD de A y B.



1.1.1. Pseudo Código:

Algoritmo 1 de Euclides

Entrada: Valores a y b pertenecientes a un dominio euclídeo

Salida: Un máximo común divisor de a y b

1. $r_0 \leftarrow a, r_1 \leftarrow b$
2. $i \leftarrow 1$
3. **Mientras** $r_i \neq 0$ **haga lo siguiente:**
    1. $r_{i+1} \leftarrow r_{i-1} \mod r_i$
    2. $i \leftarrow i + 1$
4. **El resultado es:** r_{i-1}

1.1.2. Seguimiento numérico (a mano):

1.1.3. Implementación en C++:

```
using namespace std;

int x, y, ini, medio, fin;
int multi;
int fini[14];
int medii[14];
string secuencia[10]={ "primer", "segundo" , "tercer" , "cuarto" , "quinto"
,"sexto" , "septimo" , "octavo" , "noveno" , "X"};
void Eucli()
{
cout<<"Introduzca el primer valor:";
cin>> x;
cout<<"Introduzca el segundo valor:";
cin>>y;
if(x>=y)
{
ini=x;
medio=y;
multi=ini/medio;
fin=ini%medio;
}
else
{
ini=y;
```

```
medio=x;
multi=ini/medio;
fin=ini%medio;
}
}
void cambios()
{
int cuenta;
int nule=0;
for(cuenta=0; cuenta <15; cuenta++)
{
fini[cuenta]= fin;
medii[cuenta]= medio;
if (cuenta>=10)
{
nule++;
};
cout<<"Este es el "<< secuencia[cuenta-nule]<< " paso.\n";
cout << ini << "=" << medio << "(" << multi << ") +" << fin << "\n";

if(fin==0)
{
int variable=ini%medio;
if(variable!=0)
{
cout<<"Se han acabado los despalazamientos\n";
cout<<"El maximo comun divisor es:"<< fini[cuenta-1]<<"\n\n";
}
else if(variable==0)
{
cout<<"Se han acabado los despalazamientos\n";
cout<<"El maximo comun divisor es:"<< medii[cuenta]<<"\n\n";
}
break;
}
ini=medio;
medio=fin;
multi= ini/medio;
fin= ini%medio;
}
}
int main()
{

int opcional;
```

```
int why=0;
while (!why)

cout<<"1->Algoritmo de Euclides.\n2->Salir del programa";
cin>> opcional;
switch(opcional)
{

case 1:
{
system("cls");
cout<<"Bienvenido al algoritmo de Euclides.\n";
Eucli();
cambios();
break;
}
case 2:
{
system("cls");
cout << "Saliendo del programa";
why++;
break;
}
}
}
}
```

1.2. Algoritmo extendido de Euclides:

El algoritmo de Euclides no sólo sirve para encontrar el máximo común divisor de dos números, sino que también nos indica que el mcd se puede expresar como combinación lineal de los mismos.

$$mcd(a, b) = ax + by$$

Lo que buscamos con el algoritmo extendido de Euclides, son los coeficientes de esta combinación lineal (los coeficientes de Bezout). Los cuales según la ecuación son x y ***y.***

$$\text{mcd}(a_1, b_1) = a_1x_1 + b_1y_1 \ (0<a_1<b_1)$$

$$\text{mcd}(a_2, b_2) = a_2x_2 + b_2y_2 \ (0<a_2<b_2)$$

$$\text{mcd}(a_3, b_3) = a_3x_3 + b_3y_3 \ (0<a_3<b_3)$$

...

$$\text{mcd}(a_n, b_n) = a_nx_n + b_ny_n \ (0<a_n<b_n)$$

Ejemplos:

mcd (148, 40)

148 = 3 * 40 + 28

40 = 1 * 28 + 12

28 = 2 * 12 + 4

12 = 3 * 4 + 0

mcd (148, 40) = 4

mcd (385, 78)

385 = 4 * 78+ 73

78 = 1 * 73 + 5

73 = 14 * 5 + 3

5 = 1* 3 + 2

3 = 1* 2 + 1

2 = 2 * 1 + 0

mcd (385, 78) = 1

Cuando a y b son primos (c, mcd=1), podemos tomar a los coeficientes como los inversos multiplicativos respectivamente.

1.2.1. Pseudo Código:

Entrada: $a, b \in Z^+$, *con* $a \geq b$
Salida : $d = mcd(a, b)$
SI $(b = 0)$
  $d \leftarrow a;\quad x \leftarrow 1;\quad y \leftarrow 0;$
FIN SI
retornar (d, x, y)
$x_1 \leftarrow 0;\quad x_2 \leftarrow 1;$
$y_1 \leftarrow 1;\quad y_2 \leftarrow 0;$
MIENTRAS $(b > 0)$
  $q \leftarrow \lfloor \frac{a}{b} \rfloor;\quad r \leftarrow a - qb;$
  $x \leftarrow x_2 - qx_1;\quad y \leftarrow y_2 - qy_1;$
  $a \leftarrow b;\quad b \leftarrow r;$
  $x_2 \leftarrow x_1;\ x_1 \leftarrow x;$
  $y_2 \leftarrow y_1;\quad y_1 \leftarrow y;$
FIN MIENTRAS
$d \leftarrow a;\quad x \leftarrow x_2;\quad y \leftarrow y_2;$
retornar (d, x, y)

1.2.2. Seguimiento numérico (a mano):

1.2.3. Implementación en C++:

```
int euclides_ext(int a, int b) {

int r1 = a, r2 = b;
int s1 = 1, s2 = 0;
int t1 = 0, t2 = 1;
int r, s, t;
while (r2 > 0) {
```

```
    int q = r1 / r2;
    r = r1 - (q * r2);
    r1 = r2;
    r2 = r;
    s = s1 - (q * s2);
    s1 = s2;
    s2 = s;
    t = t1 - (q * t2);
    t1 = t2;
    t2 = t;
  }
  return s1;}
```

II. Referencias Bibliográficas y web grafía

https://pier.guillen.com.mx/algorithms/05-aritmetica/05.3-euclides_ext.htm

https://euclides.org/algoritmo-de-euclides/

Algoritmo de Euclides - Wikipedia, la enciclopedia libre

https://www.lawebdelprogramador.com/codigo/Dev-C/5011-Teorema-de-Euclides-en-C-MCD.html