|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Ciencia de la Computación | | | |
| Análisis de la mejor implementación del algoritmo de Euclides | | | |
| ÁLGEBRA ABSTRACTA | | | |
|  | | | |
|  | | Eliazar Kenyi Arpasi Llanos  Rafael Alonso David Peñalva | |
|  | | 3° SEMESTRE | |
|  | | 2019 | |
|  | | | |
|  | | | |
| “El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo” | | | |
|  | | | |
|  |  | |  |
|  |  | |  |
| FIRMA | | | |

Al analizar las diferentes implementaciones del algoritmo de Euclides, se notó diferencias grandes y pequeñas entre las 7 implementaciones, a continuación, se muestra un cuadro que señala la cantidad de segundos que se demora en tiempo real (esto puede variar según los datos que se mandar por parámetro).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | algoritmo 1 | | | | algoritmo 2 | | | | algoritmo 3 | | | | algoritmo 4 | | | |
|  | a par y b impar | a impar y b impar | a impar y b par | a par y b par | a par y b impar | a impar y b impar | a impar y b par | a par y b par | a par y b impar | a impar y b impar | a impar y b par | a par y b par | a par y b impar | a impar y b impar | a impar y b par | a par y b par |
| 5 cifras | 4 | 4 | 6 | 6 | 3 | 2 | 8 | 7 | 3 | 3 | 8 | 8 | 4 | 4 | 4 | 4 |
| 10 cifras | 4 | 4 | 2 | 3 | 5 | 4 | 2 | 4 | 5 | 5 | 3 | 4 | 4 | 0 | 0 | 0 |
| 15 cifras | 2 | 3 | 5 | 4 | 3 | 6 | 6 | 5 | 3 | 6 | 6 | 6 | 0 | 0 | 0 | 0 |
| 20 cifras | 4 | 2 | 4 | 3 | 5 | 3 | 5 | 4 | 5 | 3 | 6 | 4 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | algoritmo 5 | | | | algoritmo 6 | | | | algoritmo 7 | | | |
|  | a par y b impar | a impar y b impar | a impar y b par | a par y b par | a par y b impar | a impar y b impar | a impar y b par | a par y b par | a par y b impar | a impar y b impar | a impar y b par | a par y b par |
| 5 cifras | 2 | 8 | 2 | 3 | 2 | 8 | 2 | 4 | 2 | 9 | 3 | 4 |
| 10 cifras | 3 | 2 | 4 | 8 | 3 | 2 | 4 | 9 | 4 | 3 | 5 | 10 |
| 15 cifras | 3 | 8 | 3 | 4 | 5 | 4 | 4 | 11 | 5 | 4 | 4 | 12 |
| 20 cifras | 4 | 4 | 6 | 7 | 7 | 7 | 8 | 7 | 7 | 7 | 8 | 7 |

Como se puede observar en los cuadros, el algoritmo numero 4 es el más rápido en tiempo real, a continuación, lo vamos a analizar.

unsigned long long alg\_4(unsigned long long a, unsigned long long b)

{

//SI A > B RETORNAMOS FUNCION(A,B) -> FUNCION(B,A)

if (labs(b) > labs(a))

{

return alg\_4(b, a);

}

//SI B ES IGUAL A 0 -> RETORNAMOS A

if (b == 0)

{

return a;

}

//SI A Y B % 2 == 0 ENTONCES RETORNAMOS FUNCION(A/2,B/2) \* 2

if ((a % 2 == 0) && (b % 2 == 0))

{

return 2 \* alg\_4(a / 2, b / 2);

}

//SI A % 2 == 0 Y B %2 != 0 RETORNAMOS FUCNION(A/2,B)

if ((a % 2 == 0) && (b % 2 != 0))

{

return alg\_4(a / 2, b);

}

//SI A%2 != Y B%2 == 0 ENTONCES RETORNAMOS FUNCION(A,B/2)

if ((b % 2 == 0) && (a % 2 != 0))

{

return alg\_4(a, b / 2);

}

//SI NO SE CUMPLEN LAS DEMAS CONDICIONES

//REOTRNAMOS FUNCION((A-B)/2,B)

else

{

alg\_4((labs(a) - labs(b)) / 2, b);

}

}

Se muestra una función recursiva que es reducida a la mitad uno o los dos parámetros según sea el caso. Esta implementación no tiene vueltas que conllevan a un mayor costo computacional, es implementación recursiva es muy eficaz.

En conclusión esta es la mejor implementación para el algoritmo de Euclides, aprovecha la recursividad y la menor cantidad de parámetros así como que son del mismo tipo.