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

Para la realización de los temas de investigación deberán presentar:

- Explicación detallada de cada algoritmo.
- La implementación del algoritmo.
- Detallar como el contenido de las variables cambia en cada paso (Seguimiento de código con datos de entrada reales (números enteros)).
- Comparación de los algoritmos de acuerdo a la convergencia (quién tiene el menor tiempo para llegar a los resultados) y eficiencia.
- En el informe deberán resaltar el costo computacional, el uso de memoria y el **fundamento matemático** sobre el que se sustenta los algoritmos (Como referencia, ver el formato de Informe en Anexo 1)
- Exposición. Duración: máximo 15 minutos. Como referencia, ver el formato de la presentación de las diapositivas en Anexo 2)

FECHA DE ENTREGA Y EXPOSICIONES: 11 de Junio (Grupos máximo 4 integrantes)

Instrucciones: Deberán subir al "Aula Virtual" un trabajo por grupo (responsable) y deben colocar el link al github en la parte de comentarios.

1. **Implementación de algoritmos eficientes para encontrar la exponenciación modular para exponentes y bases grandes que apliquen al RSA(8, 16, 256, 1024, 2048, etc. bits)..**
   - Exponenciación modular rápida.
   - Exponenciación modular binaria
   - Right-to-left binary exponentiation, Left-to-right k-ary exponentiation, left-to-right k-ary exponentiation
   - Teorema del resto chino
   - NaiveExponentiation
   - Uso de los teoremas Fermat y Euler
   - Otros algoritmos que sugiera
   - Comparación de los algoritmos

Referencias bibliográficas:

[01] Handbook of Applied Cryptography, Menezes, Oorschot, Vanstone. CRC Press, New York, fifth edition (2001). http://cacr.uwaterloo.ca/hac/about/chap14.pdf

[02] Chapter 10. Number theory and Cryptography. https://silo.tips/download/chapter-number-theory-and-cryptography-contents

[03] Introducción a la Teoría de Números. Ejemplos y algoritmos. Walter Mora. Capítulo 4 https://repositoriotec.tec.ac.cr/bitstream/handle/2238/6299/introducci%C3%B3n-teor%C3%ADa-n%C3%BAmeros.pdf?sequence=1&isAllowed=y

Otros algoritmos:

```
Exponenciación(a,p,n)
    Entrada: enteros a, p, n
    Salida: r=a^p mod n
    r=1
    For i=1 to p do
            r=(r.a) mod n
    Return r
```

```
Exponenciación(a,p,n)
    Entrada: enteros a, p, n
    Salida: r=a^p mod n

    If p=0 then
            Return 1
    If p es par then
            t=Exponenciación(a, p/2,n)
            Return  t^2 mod n
    t=exponenciación (a, (p-1)/2,n)
    Return a(t^2 mod n) mod n
```

Input. Positive integers N, g, and A.

1. Set $a = g$ and $b = 1$.

2. Loop while $A > 0$.

3. If $A \equiv 1 \pmod 2$, set $b = b \cdot a \pmod N$.

4. Set $a = a^2 \pmod N$ and $A = \lfloor A/2 \rfloor$.

5. If $A > 0$, continue with loop at Step **2**.

6. Return the number b, which equals $g^A \pmod N$.

INPUT: $a \in \mathbb{Z}_n$, and integer $0 \leq k < n$ whose binary representation is $k = \sum_{i=0}^{t} k_i 2^i$.
OUTPUT: $a^k \bmod n$.

1. Set $b \leftarrow 1$. If $k = 0$ then return(b).
2. Set $A \leftarrow a$.
3. If $k_0 = 1$ then set $b \leftarrow a$.
4. For i from 1 to t do the following:
   4.1 Set $A \leftarrow A^2 \bmod n$.
   4.2 If $k_i = 1$ then set $b \leftarrow A \cdot b \bmod n$.
5. Return(b).

Anexo 1

TRABAJO DE INVESTIGACION E IMPLEMENTACION DE ALGORITMOS DE EXPONENCIACIÓN RÁPIDA

Nombres y Apellidos de los Integrantes del grupo, describiendo el aporte que realizaron en el trabajo (Subir Responsable del grupo. ***Además adicionar el informe al github***)

- **Resumen**
  Pequeña descripción de que algoritmos analizó, que criterios de evaluación tuvieron en cuenta y que algoritmo (s) tuvieron mejor desempeño
- **Introducción**
  En la redacción resaltar el problema y objetivos de la investigación.
- **Contenido Teórico**
  Para cada algoritmo describir:
  . El pseudo- algoritmo
  . Seguimiento numérico (a mano)
  . Implementación en C++

- **Análisis de los algoritmos**

  . Describir las características del procesador y sistema operativo en que están evaluando los algoritmos
  . Comparar los diferentes algoritmos por tiempo de ejecución vs. Nro. de Bits (Graficar resultados). Tiempo de Ejecución con diferente número de bits (128 – 512 - 1024 - 2046)

  . Comparar algoritmos por el cálculo computacional

- **Conclusiones Generales**
- **Referencias**

Anexo 2

Contenido Diapositivas

- **Introducción**
  Enumerar los algoritmos que utilizó (tanto para el algoritmo de Euclides y algoritmo extendido de euclides)
- **El mejor Algoritmo de Exponenciación modular**
  . Explicar el algoritmo e Implementación
  . Mostrar la comparación con el resto de algoritmos que justifique que sea el mejor.

- **Conclusiones**
- **Referencias**