EL TEOREMA GENERAL
DE LA FACTORIZACIÓN DE CORDERO
ALGORITMOS Y PROGRAMAS COMPUTACIONALES.
CUARTO TEOREMA DE LA FACTORIZACIÓN DE CORDERO

Universidad Internacional San Isidro Labrador. Costa Rica.

Máster: Ronald Cordero Méndez.

Ronald.come@gmail.com

Bioinformático: Roberto Reinosa Fernández

Roberto117343@gmail.com

RESUMEN

El artículo se centra en la investigación del Teorema General de La Factorización de Cordero, el Cuarto Teorema de la Factorización de Cordero en el Conjunto de Los Números Enteros, los Algoritmos de Cordero y los software $R^3 * C$ construidos a partir de estos Algoritmos. El Teorema General de La Factorización de Cordero permite factorizar en dos factores, los números polinomiales de la forma $r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n + 1)r + 1$, con $r \epsilon Z, n \epsilon Q, r \neq 0$ y $p \epsilon \{2, 3, 5, 11, 17, 41\}$. El Cuarto Teorema de La Factorización de Cordero en El Conjunto de los Números Enteros obtiene la factorización en dos factores de los números polinomiales de la forma $2n^2 + p \; con \; n \epsilon Z$ y $p \epsilon \{3, 5, 11, 29\}$. Los Algoritmos de Cordero se utilizan para factorizar en sus factores primos estos números polinomiales y los software construidos a partir del Teorema y los algoritmos nos permiten verificar tales descubrimientos matemáticos.

PALABRAS CLAVES.

Factorización, Números enteros, Algoritmos, Programas computacionales, Software, Teorema

ABSTRACT

The article focuses on the investigation of the General Theorem of Cordero Factorization, and the Fourth Theorem of Cordero Factorization on the Set of Integers, the Cordero Algorithms and the $R^3 * C$ software built from these Algorithms. Cordero's General Factorization Theorem allows you to factor polynomial numbers of the form $r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n + 1)r + 1$, con $r \epsilon Z, n \epsilon Q, r \neq 0$ y $p \epsilon \{2, 3, 5, 11, 17, 41\}$. The Fourth Theorem of Cordero Factorization on the Set of Integers obtains the factorization into two factors of the polynomial numbers of the form $2n^2 + p \; con \; n \epsilon Z$ y $p \epsilon \{3, 5, 11, 29\}$. Cordero's Algorithms are used to factorize these polynomial numbers into their prime factors and the software built from the Theorem and the algorithms allow us to verify such mathematical discoveries.

KEYWORDS.

Factorization, Integers, Algorithms, Computer programs, Software, Theorem.

1. INTRODUCCIÓN.

El objetivo principal de los matemáticos es resolver problemas y descubrir patrones, como es el caso de encontrar una forma matemática de descomponer números enteros en sus factores primos. Por lo que tomamos las palabras de Gauss cuando dijo que el honor de la Matemática requería que resolviéramos un problema tan elegante como la factorización de los números enteros. Sabemos que encontrar una solución definitiva para la factorización completa de números enteros no es fácil, ni lo será. Pero se han logrado avances que nos permite creer que posiblemente se pueda encontrar esta solución tan esperada.

Hace unos años parecía que solo a los matemáticos les interesaba la factorización de números enteros, pero recientemente personas que no son matemáticos les interesa el tema, por ejemplo, a los profesionales en computación y a los que requieren de seguridad informática. Pero a pesar del aumento de la capacidad de los computadores y de la creación de diferentes algoritmos para mejorar el tiempo que requiere la factorización de un número entero, el problema de la factorización es un problema sin resolver hasta el momento. No hay métodos sencillos para obtener la factorización completa de un número entero.

Se considera un algoritmo eficiente si pudiera resolver la factorización de un número entero en tiempo polinomial, por lo que los matemáticos y profesionales en otras áreas de todo el mundo se esmeran por encontrar una solución a tan difícil problema matemático y computacional.

A pesar de los descubrimientos matemáticos y el aumento de potencia de cómputo, diferentes algoritmos utilizados y el respaldo de la computación distribuida, si se habla de factorizar enteros de centenas y más aún de miles de dígitos decimales, hoy día es computacionalmente ineficiente, debido por un lado por la aleatoriedad de la distribución de los números primos y por el otro a la dificultad de factorizar números muy grandes de factores primos grandes.

2. TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO.

El teorema que se publica en este artículo tiene como objetivo factorizar números enteros de la forma $r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1$, con $r \in Z,\ n \in Q,\ r \neq 0$ y $p \in \{2,3,5,11,17,41\}$.

El Teorema General de la Factorización de Cordero está publicado en la Revista el Labrador de la Universidad Internacional San Isidro Labrador, https://revistaellabrador.net/index.php/RevistaElLabrador/article/view/89

También está publicado a través de videos, como https://youtu.be/jnpnTrN7hh4

TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO.

Sea $r, s, x, T \epsilon Z,\ s \neq 0, r \neq 0$ y $p \epsilon \{2,3,5,11,17,41\}$.

I. Si

$$\boldsymbol{n = \frac{s((r-1)s+r)x^2 + \left((r-1)s^2 - (r-2)s - r\right)x + s((r-1)s+r)p - (r-1)s}{r} + [s^2x^2 + s(s-2)x + ps^2 - s + 1] * T}$$

entonces $r^2n^2 + r(r-2)n + pr^2 - r + 1$ es compuesto y dos de sus factores tienen la forma:

$$f_1 = s^2x^2 + s(s-2)x + ps^2 - s + 1$$

$$f_2 = f_1r^2T^2 + r(2\beta + r - 2)T + \left((r-1)s + r\right)^2 x^2 + \left((r-1)s + r\right)\left((r-1)s - r + 2\right)x + p\left((r-1)s + r\right)^2 - (r-1)\left((r-1)s + r\right) + (r-1)^2$$

donde $\beta = s\left((r-1)s + r\right)x^2 + \left((r-1)s^2 - (r-2)s - r\right)x + s\left((r-1)s + r\right)p - (r-1)s$

II. Si

$$n = \frac{x^2 + (r-1)x + p - (r-1)}{r} + (x^2 - x + p) * T$$

entonces $r^2n^2 + r * (r-2)n + pr^2 - r + 1$ es compuesto y dos de sus factores tienen la forma:

$$f_1 = x^2 - x + p$$

$$f_2 = f_1r^2T^2 + r(2\beta + r - 2)T + x^2 + (2r-1)x + r^2 - r + p$$

donde $\beta = x^2 + (r-1)x + p - (r-1)$

Nota: La parte II del Teorema ha sido modificada por el Matemático: Ronald Cordero Méndez.

2.1.1 Aplicación de la primera parte del Teorema.

Sea $r = -5,\ p = 17,\ s = 9,\ T = 3,\ x = -7$

$$n = \frac{s\left((r-1)s + r\right)x^2 + \left((r-1)s^2 - (r-2)s - r\right)x + s\left((r-1)s + r\right)p - (r-1)s}{r} + [s^2x^2 + s(s-2)x + ps^2 - s + 1] * T$$

$$n = \frac{-531x^2 - 418x - 8973}{-5} + [81x^2 + 63x + 1369] * 3$$

$$n = \frac{32066}{5} + 4897 * 3$$
$$n = \frac{105521}{5}$$

El polinomio a evaluar es:

$p(n) = r^2n^2 + r * (r-2)n + pr^2 - r + 1 = 25n^2 + 35n + 431$

$$p\left(\frac{105521}{5}\right) = 25\left(\frac{105521}{5}\right)^2 + 35 * \left(\frac{105521}{5}\right) + 431 = 11135420519$$
$$= 4897 * 2273927$$

El factor 2273927 se puede encontrar por división (como ya conocemos el factor 4897 puesto que es parte de n). Pero también hay una fórmula para encontrarlo, pero por su dificultad es poco utilizada).

La fórmula para encontrar el factor f_2 es:

$$f_2 = f_1r^2T^2 + r(2\beta + r - 2)T + \big((r-1)s + r\big)^2x^2 + \big((r-1)s + r\big)\big((r-1)s - r + 2\big)x + p\big((r-1)s + r\big)^2 - (r-1)\big((r-1)s + r\big) + (r-1)^2$$

$$f_2 = 4897 * 25 * 9 - 5(2 * (-32066) - 5 - 2) * 3 + 3481 * 49 - 59 * (-47) * (-7) + 17 * 3481 + 6 * (-59) + 36 = 2273927$$

Además el f_2 se puede encontrar utilizando la fórmula $f_2 = \frac{(4p-1)*{r_*}^2 + M^2}{4}$ donde

$r_* = rs(T+1) + r - s$ y $M = 2xr_* + (s-2) * (rT + r - 1) + r$

En efecto:

$$r_* = -5 * 9 * 4 - 5 - 9 = -194$$

$$M = 2 * (-7) * (-194) + 7 * (-5 * 3 - 5 - 1) - 5 = 2564$$

$$f_2 = \frac{(4p-1) * {r_*}^2 + M^2}{4} = \frac{67 * 194^2 + 2564^2}{4} = 2273927$$

2.1.2 Aplicación de la segunda parte del Teorema.

Sea $r = 12,\ p = 11, x = -8, T = 6$

$$n = \frac{x^2 + (r-1)x + p - (r-1)}{r} + (x^2 - x + p) * T$$

$$n = \frac{(-8)^2 + 11 * (-8) + 11 - (11)}{12} + (8^2 + 8 + 11) * 6$$

$$n = -2 + 83 * 6$$

$$n = 496$$

Entonces $r^2n^2 + r * (r-2)n + pr^2 - r + 1 =$

$144 * 496^2 + 12 * 10 * 496 + 11 * 144 - 12 + 1 = 35487397 = 83 * 427559$

El factor 427559 se calculó por división, pero también se puede calcular con la fórmula $f_2 = f_1r^2T^2 + r(2\beta + r - 2)T + x^2 + (2r-1)x + r^2 - r + p =$

$$f_2 = 83 * 144 * 36 + 12 * (2 * (-24) + 12 - 2) * 6 + 8^2 - (2 * 12 - 1) * 8 + 144 - 12 + 11$$

$$f_2 = 427559$$

3. ALGORITMO DE CORDERO PARA FACTORIZAR COMPLETAMENTE NÚMEROS DE LA FORMA $\boldsymbol{r^2n^2 + r(r-2)n + pr^2 - r + 1}$

El algoritmo permite factorizar números polinomiales. El usuario da valores de entrada r, n , p y el algoritmo construye el número polinomial $r^2n^2 + r(r-2)n + pr^2 - r + 1$, luego lo factoriza en sus factores primos. Pero también se puede utilizar el algoritmo para factorizar números polinomiales obtenidos del Teorema General de La Factorización de Cordero. Partimos de los valores de $r, s, T\ y\ x$ construimos el valor de n y luego aplicamos el algoritmo r, n, p.

Algoritmo r, n, p (Algoritmo de Cordero)

Sea $n \epsilon Z, r \epsilon Z, r \neq 0,\ p \epsilon \{2, 3, 5, 11, 17, 41\}\ y\ T_{\emptyset} \epsilon Z,\ T_{\emptyset} \neq 0$

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2(r + 2rn - 2)T_{\emptyset} + r^2}$$

I. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces $r^2n^2 + r(r-2)n + pr^2 - r + 1$ es un número compuesto, caso contrario $r^2n^2 + r(r-2)n + pr^2 - r + 1$ es un número primo.

II. Si $r^2n^2 + r(r-2)n + pr^2 - r + 1$ es un número compuesto con $(T_{\phi}, \phi), T_{\phi} \epsilon Z$, $\phi \epsilon Z\ y\ \frac{-r + T_{\emptyset} \pm \phi}{2 * T_{\phi}} = \frac{t}{b}$ con $t\ y\ b$ primos entre sí entonces $t^2 - tb + pb^2$ es un factor de $r^2n^2 + r(r-2)n + pr^2 - r + 1$.

Nota: El valor de n puede pertenecer a Q, pero debe ser generado por el Teorema General de La Factorización de Cordero.

3.1 APLICACIONES DEL ALGORITMO DE CORDERO r, n, p.

En este apartado analizaremos la aplicabilidad del algoritmo, su importancia en la factorización completa de un número entero polinomial de la forma

$r^2n^2 + r(r-2)n + pr^2 - r + 1$, además de sus limitaciones.

Aplicación 1.

Sea $r = 7, n = 786\ y\ p = 41$

Luego:

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2(r + 2rn - 2)T_{\emptyset} + r^2}$$

$$\emptyset = \sqrt{-163 * {T_{\emptyset}}^2 + 22018 * T_{\emptyset} + 49}\ , T_{\emptyset} \epsilon Z\ , T_{\emptyset} \neq 0$$

Si aplicamos el algoritmo, obtenemos un único par ordenado, o sea el número polinomial es biprimo o semiprimo:

$$T_{\emptyset} = 1 \quad y \quad \emptyset = 148$$

Par ordenado: $(1, 148)$

Y el número polinomial es:

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = 7^2 * 786^2 + 7 * 5 * 786 + 41 * 49 - 7 + 1 = 30301517$$

Así:

$$\frac{\boldsymbol{-r + T_{\emptyset} + \phi}}{\boldsymbol{2 * T_{\phi}}} = \frac{\boldsymbol{-7 + 1 + 148}}{\boldsymbol{2}} = \boldsymbol{71}$$

Por lo que: $71^2 - 71 + 41 = 5011$ es un factor del número polinomial.

$$\frac{\boldsymbol{-r + T_{\emptyset} - \phi}}{\boldsymbol{2 * T_{\phi}}} = \frac{\boldsymbol{-7 + 1 - 148}}{\boldsymbol{2}} = \boldsymbol{-77}$$

Por lo que: $77^2 + 77 + 41 = 6047$ es el otro factor del número polinomial.

Como no hay más pares ordenados enteros generados por el algoritmo, tenemos:

$$30301517 = 5011 * 6047$$

Aplicación 2.

Continuando con el ejemplo de aplicación del Teorema General de La Factorización de Cordero.

Sea $r = -5,\ n = \frac{105521}{5} \quad y \quad p = 17$

Luego:

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2(r + 2rn - 2)T_{\emptyset} + r^2}$$

$$\emptyset = \sqrt{-67 * {T_{\emptyset}}^2 - 422098 * T_{\emptyset} + 25}\ , T_{\emptyset} \epsilon Z\ , T_{\emptyset} \neq 0$$

Si aplicamos el algoritmo, obtenemos:

$$(1746, 23081) \qquad (1788, 23249) \quad y \quad (5086, 20339)$$

Y el número polinomial es:

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = 11135420519 = 4897 * 2273927$$

Para $(1746, 23081)$

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}+\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+1746+23081}}{\mathbf{2*1746}} = \frac{\mathbf{64}}{\mathbf{9}}$$

Por lo que: $64^2 - 64*9 + 17*81 = 4897$ es un factor del número polinomial.

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}-\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+1746-23081}}{\mathbf{2*1746}} = -\frac{\mathbf{1185}}{\mathbf{194}}$$

Por lo que: $1185^2 + 1185*194 + 17*194^2 = 2273927$ es el otro factor del número polinomial y que son precisamente los factores que se obtienen con el Teorema.

Para $(1788, 23249)$ obtenemos:

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}+\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+1788+23249}}{\mathbf{2*1788}} = \frac{\mathbf{12521}}{\mathbf{1788}}$$

Por lo que: $12521^2 - 12521*1788 + 17*1788^2 = 188735941$ es un factor del número polinomial.

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}-\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+1788-23249}}{\mathbf{2*1788}} = \mathbf{-6}$$

Por lo que: $6^2 + 6 + 17 = 59$ es otro factor del número polinomial.

Para $(5086, 20339)$ obtenemos:

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}+\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+5086+20339}}{\mathbf{2*5086}} = \frac{\mathbf{5}}{\mathbf{2}}$$

Por lo que: $5^2 - 5*2 + 17*2^2 = 83$ es un factor del número polinomial.

$$\frac{-\boldsymbol{r}\ +\boldsymbol{T}_{\emptyset}-\boldsymbol{\phi}}{\mathbf{2}*T_{\phi}} = \frac{\mathbf{5+5086-20339}}{\mathbf{2*5086}} = -\frac{\mathbf{3812}}{\mathbf{2543}}$$

Por lo que: $3812^2 + 3812*2543 + 17*2543^2 = 134161693$ es otro factor del número polinomial.

Como no hay más pares ordenados enteros generados por el algoritmo, tenemos:

$$11135420519 = 59 * 83 * 2273927$$

Aplicación 3.

Sea $r = 8,\ x = 12,\ s = 15\ ,\ T = 4\ \ y\ \ p = 41$

Utilizando el Teorema General de La Factorización de Cordero se construye el valor de n.

$$\boldsymbol{n = \frac{s\big((r-1)s+r\big)x^2 + \Big((r-1)s^2-(r-2)s-r\Big)x + s\big((r-1)s+r\big)p-(r-1)s}{r} + [s^2x^2+s(s-2)x+ps^2-s+1]*T}$$

$$\boldsymbol{n = \frac{1695x^2+1477x+69390}{8} + [225x^2+195x+9211]*4}$$

$$\boldsymbol{n = \frac{165597}{4} + [43951]*4}$$

$$\boldsymbol{n = \frac{868813}{4}}$$

Luego:

$$\emptyset = \sqrt{(1-4p){T_\emptyset}^2 + 2(r+2rn-2)T_\emptyset + r^2}$$

$$\emptyset = \sqrt{-163*{T_\emptyset}^2 + 6950516*T_\emptyset + 64}\ ,T_\emptyset \epsilon Z\ ,T_\emptyset \neq 0$$

Si aplicamos el algoritmo, obtenemos:

$$(8895, 221197)\ \ y\ \ \ (12057, 245167)$$

Y el número polinomial es:

$$\begin{aligned} &r^2n^2 + r(r-2)n + pr^2 - r + 1 \\ &\qquad = 64*\left(\frac{868813}{4}\right)^2 + 8*6*\frac{868813}{4} + 41*64 - 8 + 1 \\ &\qquad = 3019354544249 \end{aligned}$$

Para $(8895, 221197)$

$$\frac{\boldsymbol{-r\ +T_\emptyset + \phi}}{\boldsymbol{2}*T_\phi} = \boldsymbol{\frac{-8+8895+221197}{2*8895} = \frac{194}{15}}$$

Por lo que: $194^2 - 194*15 + 41*15^2 = 43951$ es un factor del número polinomial.

5

$$\frac{\boldsymbol{-r\ +T_\emptyset - \phi}}{\boldsymbol{2}*T_\phi} = \boldsymbol{\frac{-8+8895-221197}{2*8895} = -\frac{77077}{593}}$$

Por lo que: $77077^2 + 77077 * 593 + 41 * 593^2 = 6000988199$ es otro factor del número polinomial.

Para $(12057, 245167)$ obtenemos:

$$\frac{\boldsymbol{-r + T_{\emptyset} + \phi}}{\boldsymbol{2 * T_{\phi}}} = \frac{\mathbf{-8 + 12057 + 245167}}{\mathbf{2 * 12057}} = \frac{\mathbf{32}}{\mathbf{3}}$$

Por lo que: $32^2 - 32 * 3 + 41 * 3^2 = 1297$ es un factor del número polinomial.

$$\frac{\boldsymbol{-r + T_{\emptyset} - \phi}}{\boldsymbol{2 * T_{\phi}}} = \frac{\mathbf{-8 + 12057 - 245167}}{\mathbf{2 * 12057}} = \frac{\mathbf{-38853}}{\mathbf{4019}}$$

Por lo que: $38853^2 + 38853 * 4019 + 41 * 4019^2 = 2327952617$ es otro factor del número polinomial.

Luego:

$$3019354544249 = 1297 * 43951 * 52967$$

El último factor se obtiene por división, y esta es la factorización prima del número polinomial.

4. ALGORITMOS PARA FACTORIZAR LOS DOS FACTORES QUE SE OBTIENEN CON EL TEOREMA GENERAL DE LA FACTORIZACIÓN DE CORDERO. FACTORES F_1 Y F_2. UN ALGORITMO PARA CADA FACTOR.

Cuando utilizamos el Teorema General de la Factorización de Cordero obtenemos dos factores del número polinomial, pero estos dos factores generalmente no conforman la factorización prima del número polinomial, por lo que utilizaremos los siguientes algoritmos para encontrar su factorización completa o factorización prima.

Algoritmos para f_1 y f_2 (Algoritmos de Cordero)

Sea $r, x, s, T \epsilon Z, r \neq 0, s \neq 0$ y $p \epsilon \{2,3,5,11,17,41\}$ $además$ $T_{\emptyset} \epsilon Z, T_{\emptyset} \neq 0$

Algoritmo para f_1

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2(s + 2sx - 2)T_{\emptyset} + s^2}$$

$$f_1 = s^2 * x^2 + s(s-2) * x + ps^2 - s + 1$$

I. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_1 es un número compuesto, caso contrario f_1 es un número primo.

II. Si f_1 es un número compuesto con $(T_{\phi}, \phi), T_{\phi} \epsilon Z, \phi \epsilon Z$ y $\frac{-s + T_{\emptyset} \pm \phi}{2 * T_{\phi}} = \frac{t}{b}$ con t y b primos entre sí entonces $t^2 - tb + pb^2$ es un factor de f_1.

Algoritmo para f_2

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2M\,T_{\emptyset} + {r_*}^2}$$

Donde $r_* = r * s * (T+1) + r - s$ y $M = 2 * x * r_* + (s-2)(r * T + r - 1) + r$

$$f_2 = \frac{(4p-1){r_*}^2 + M^2}{4}$$

III. Si existe al menos un $T_{\phi} \epsilon Z, T_{\emptyset} \neq 0$ tal que $\phi \epsilon N$ entonces f_2 es un número compuesto, caso contrario f_2 es un número primo.

IV. Si f_2 es un número compuesto con $(T_{\phi}, \phi), T_{\phi} \epsilon Z, \phi \epsilon Z$ y $\frac{-r_* + T_{\emptyset} \pm \phi}{2 * T_{\phi}} = \frac{t}{b}$ con t y b primos entre sí entonces $t^2 - tb + pb^2$ es un factor de f_2.

4.1 Aplicaciones de los Algoritmos de Cordero para f_1 y f_2.

Aplicación 1.

Sea $r = 19,\ x = -133, T = 15,\ s = 48\ y\, p = 41$

$$\boldsymbol{n = \frac{s\big((r-1)s + r\big)x^2 + \Big((r-1)s^2 - (r-2)s - r\Big)x + s\big((r-1)s + r\big)p - (r-1)s}{r} + [s^2x^2 + s(s-2)x + ps^2 - s + 1] * T}$$

$$\boldsymbol{n = \frac{42384\,x^2 + 40637x + 1736880}{19} + [2304\,x^2 + 2208\,x + 94417] * 15}$$

$$\boldsymbol{n} = \frac{\mathbf{746062735}}{\mathbf{19}} + \mathbf{40556209} * \mathbf{15}$$

$$\boldsymbol{n} = \frac{\mathbf{12304582300}}{\mathbf{19}}$$

El número polinomial a factorizar es:

$$r^2n^2 + r(r-2)n + pr^2 - r + 1$$
$$= 19^2 * \left(\frac{\mathbf{12304582300}}{\mathbf{19}}\right)^2 + 19 * 17 * \frac{\mathbf{12304582300}}{\mathbf{19}} + \mathbf{41} * \mathbf{19^2} - \mathbf{19} + \mathbf{1}$$
$$= 151402745786651203883$$

Factorización de $\boldsymbol{f_1}$.

$$f_1 = s^2 * x^2 + s(s-2) * x + ps^2 - s + 1$$

$$f_1 = 2304 * x^2 + 2208 * x + 94417$$

$$f_1 = 2304 * 133^2 - 2208 * 133 + 94417$$

$$f_1 = 40556209$$

El algoritmo que utilizamos es:

$$\emptyset = \sqrt{(1-4p){T_\emptyset}^2 + 2(s + 2sx - 2)T_\emptyset + s^2}$$

$$\emptyset = \sqrt{-163 * {T_\emptyset}^2 - 25444 * T_\emptyset + 2304}\ , T_\emptyset \epsilon Z\ , T_\emptyset \neq 0$$

Si aplicamos el algoritmo, obtenemos:

$$T_\emptyset = -140 \quad y \quad \emptyset = 608$$

Par ordenado: $(140, 608)$

Así:

$$\frac{-s\ + T_{\emptyset} + \phi}{2 * T_{\phi}} = \frac{-48 - 140 + 608}{-280} = \frac{-3}{2}$$

Por lo que: $3^2 + 3 * 2 + 41 * 4 = 179$ es un factor del número polinomial.

$$\frac{-s\ + T_{\emptyset} - \phi}{2 * T_{\phi}} = \frac{-48 - 140 - 608}{-280} = \frac{199}{70}$$

Por lo que: $199^2 - 199 * 70 + 41 * 70^2 = 226571$ es el otro factor del número polinomial.

Como no hay más pares ordenados enteros generados por el algoritmo, tenemos que la factorización prima de f_1 es:

$$f_1 = 40556209 = 179 * 226571$$

Factorización de f_2

Calculamos primero $r_* = rs(T+1) + r - s = 14563$

Y $M = 2 * x * r_* + (s-2) * (rT + r - 1) + r = -3873758 + 13957 = -3859801$

Luego:

$$f_2 = \frac{(4p-1){r_*}^2 + M^2}{4}$$

$$f_2 = \frac{163 * 14563^2 + 3859801^2}{4} = 3733158239387$$

El algoritmo que utilizamos es:

$$\emptyset = \sqrt{(1-4p){T_{\emptyset}}^2 + 2 * M * T_{\emptyset} + {r_*}^2}$$

$$\emptyset = \sqrt{-163 * {T_{\emptyset}}^2 + 7719602 * T_{\emptyset} + 212080969}\ , T_{\emptyset} \epsilon Z\ , T_{\emptyset} \neq 0$$

Si aplicamos el algoritmo, obtenemos que no hay puntos enteros, por lo que

$f_2 = 3733158239387$ es un número primo.

Tenemos que la factorización prima del número polinomial es:

$$151402745786651203883 = 179 * 226571 * 3733158239387$$

5. SOFTWARES $R^3 * C$. BASADOS EN LOS ALGORITMOS DE CORDERO

Los Softwares o Programas Computacionales $R^3 * C$ están basados en el algoritmo de Cordero r, n, p, el Teorema General de La Factorización de Cordero y en los algoritmos de los factores f_1 y f_2.

Su nombre se deriva del nombre de sus coautores: El Costarricense Ronald Cordero Méndez y el Español Roberto Reinosa Fernández.

5.1 PROGRAMA TEST DE PRIMALIDAD r, n, p.

Su objetivo es detectar si el número polinomial

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1,$$

$r \neq 0$ y $p \in \{2, 3, 5, 11, 17, 41\}$ es un número primo o compuesto.

Puede localizar el código computacional que detecta la primalidad del número polinomial en el link:

https://github.com/roberto117343/R3C/blob/main/R3C_PRIMALIDAD.cpp

Aplicación I.

Sea $r = 34321,$ $p = 41$ y $n = 23432$

```
Introduce valor r: 34321
Introduce valor p: 41
Introduce valor n: 23432

Número Polinomial: 646780844505219313

Es primo

Tiempo transcurrido: 1.9779619150 segundos
------------------------
```

El número polinomial es

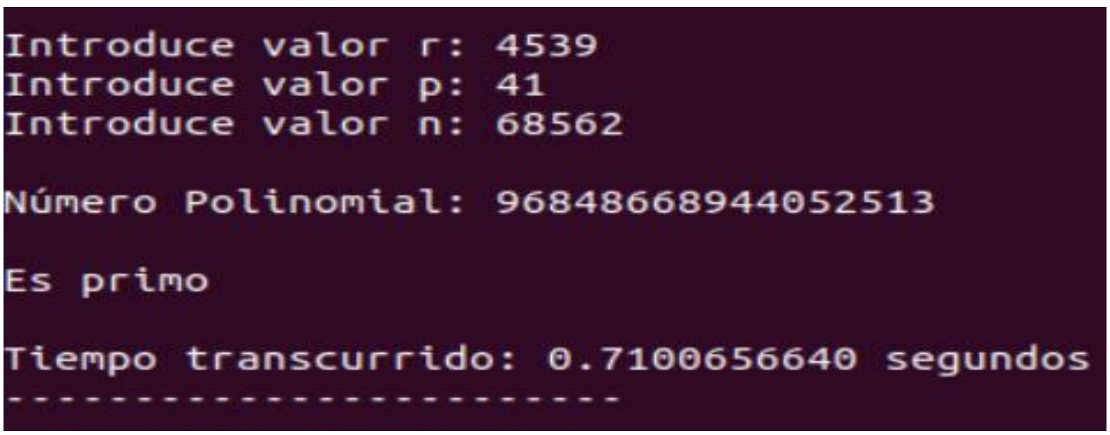
$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n + 1)r + 1$$

$$= 34321^2 * (23432^2 + 23432 + 41) - (2 * 23432 + 1) * 34321 + 1$$
$$= 646780844505219313$$

El programa necesitó de 1,977961915 segundos para detectar la primalidad del número de 18 dígitos.

Aplicación II.

Sea $r = 4539,\ p = 41\ y\ n = 68562$



El número polinomial es

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n + 1)r + 1$$

$$= 4539^2 * (68562^2 + 68562 + 41) - (2 * 68562 + 1) * 4539 + 1$$
$$= 96848668944052513$$

El programa necesitó de 0,710065664 segundos para detectar la primalidad del número de 18 dígitos.

Aplicación III.

Sea $r = 4563,\ p = 11\ y\ n = 34577$

```
Introduce valor r: 45671
Introduce valor p: 41
Introduce valor n: 56841

Número Polinomial: 6739258376500338991

Es primo

Tiempo transcurrido: 5.8419784200 segundos
-------------------------
```

El programa necesitó de 5,84197842 segundos para detectar que el número polinomial de 19 dígitos es primo.

Aplicación IV.

Sea $r = 56543,\ p = 41\ y\ n = 34321$

```
Introduce valor r: 56543
Introduce valor p: 41
Introduce valor n: 34321

Número Polinomial: 3766085965796675999

Es compuesto

Tiempo transcurrido: 2.0300584400 segundos
-------------------------
```

El programa necesitó de 2,03005844 segundos para detectar que el número polinomial de 19 dígitos es compuesto.

El algoritmo del Test de Primalidad r, n, p puede detectar si un número polinomial de la forma $r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1$

$r \neq 0\ \ y\ \ p \in \{2,3,5,11,17,41\}$ es primo o compuesto sin importar la cantidad de dígitos que posea, pero el límite para este programa está en unos 19 dígitos. Se están evaluando métodos más precisos para superar este límite, sin que aumente demasiado el tiempo de cálculo.

5.2 PROGRAMA $\boldsymbol{r, n, p}$ FACTORIZADOR DE NÚMEROS POLINOMIALES : $\boldsymbol{r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1}$, $\boldsymbol{r \neq 0 \ y \ p \in \{2, 3, 5, 11, 17, 41\}}$

Su objetivo es factorizar completamente números de la forma

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1 ,$$

$$r \neq 0 \ y \ p \in \{2,3,5,11,17,41\}$$

Puede localizar el código en el link:

https://github.com/roberto117343/R3C/blob/main/R3C_RNP.cpp

Aplicación I.

Sea $r = 6786, \ p = 41 \ y \ n = 78976$

```
Introduce valor r: 6786
Introduce valor p: 41
Introduce valor n: 78976

Número Polinomial: 287225820179111971

Es biprimo

Factor 1: 199921
Factor 2: 1436696596051

Tiempo transcurrido: 1.20720603 segundos
------------------------
```

El programa necesitó de 1, 20720603 segundos para factorizar en sus factores primos el número polinomial de 18 dígitos 287225820179111971

Aplicación II.

Sea $r = 56765, \ p = 41 \ y \ n = 23431$

```
Introduce valor r: 56765
Introduce valor p: 41
Introduce valor n: 23431

Número Polinomial: 1769137135935494231

Factor 1: 8115166841
Factor 2: 251
Factor 3: 347
Factor 4: 2503

Tiempo transcurrido: 3.01498640 segundos
------------------------
```

Para factorizar el número de 19 dígitos necesitó de 3,0149864 segundos.

Para este caso se vuelve a tener la limitación en 19 dígitos. Esto podría solucionarse utilizando métodos alternativos como el uso de las librerías Boost Multiprecision. El problema es que al utilizar estas librerías el tiempo de cómputo aumenta demasiado, ya que requiere de más tiempo para realizar cada cálculo que utilice variables de dicha librería.

5.3 PROGRAMA f_1 *y* f_2

Utiliza el Teorema General de La factorización de Cordero, y su objetivo es factorizar en sus factores primos, los números polinomiales de la forma

$n^2 + (r-2)n + pr^2 - r + 1, r \neq 0 \; y \; p \in \{2,3,5,11,17,41\}$ en dos factores $f_1 \; y \; f_2$; luego factoriza por separado a $f_1 \; y \; f_2$ utilizando un algoritmo diferente para cada factor. Por lo que su alcance en una computadora se amplía a números polinomiales de 34 dígitos. Al utilizar polinomios generadores de números primos sus factores primos siempre son grandes.

En el siguiente link encontrará el código del Software construido por el Bioinformático Roberto Reinosa Fernández y que utiliza el algoritmo para f_1, f_2 del Máster Ronald Cordero Méndez.

https://github.com/roberto117343/R3C/blob/main/R3C_F1_F2.cpp

Aplicación I.

Sea $r = 136, \; p = 41, \; s = 365, x = 655 \; y \; T = 125$

```
Introduce valor r: 136
Introduce valor p: 41
Introduce valor s: 365
Introduce valor x: 655
Introduce valor T: 125

Número Polinomial: 962330960207965767385991596441

Factor: 57249101711
Factor: 2042700047
Factor: 9781
Factor: 841333

Tiempo transcurrido: 8.29879723 segundos
------------------------
```

El número polinomial que se obtiene tiene 30 dígitos y lo factoriza en 8,29879723 segundos, por lo que es rápido en los cálculos.

Aplicación II.

Sea $r = 124,\ p = 41,\ s = 1124, x = 125\ \ y\ T = 221$

```
Introduce valor r: 124
Introduce valor p: 41
Introduce valor s: 1124
Introduce valor x: 125
Introduce valor T: 221

Número Polinomial: 301573707756480998095516056797

Factor: 421
Factor: 47386433
Factor: 15116712769005919129

Tiempo transcurrido: 7.90238151 segundos
------------------------
```

El número polinomial posee 30 dígitos y lo factoriza en 7,9 segundos aproximadamente en una computadora portátil.

Aplicación III.

Sea $r = 587,\ p = 41,\ s = 668,\ x = 225\ \ y\ T = 123$

```
Introduce valor r: 587
Introduce valor p: 41
Introduce valor s: 668
Introduce valor x: 225
Introduce valor T: 123

Número Polinomial: 2732086353924055882286611650809

Factor: 30113
Factor: 754109
Factor: 51249514791403
Factor: 2347559

Tiempo transcurrido: 22.40464673 segundos
-----------------------
```

Aplicación IV.

Sea $r = 998,\ p = 17,\ s = 525,\ x = 656\ y\ T = 23$

```
Introduce valor r: 998
Introduce valor p: 17
Introduce valor s: 525
Introduce valor x: 656
Introduce valor T: 23

Número Polinomial: 8096943126751712296917500440063

Factor: 1511
Factor: 78620891
Factor: 73
Factor: 157
Factor: 103
Factor: 5737626383961

Tiempo transcurrido: 46.98377834 segundos
-----------------------
```

El número polinomial posee 31 dígitos. En estos casos de aplicación del Teorema, los números polinomiales siempre son compuestos, por las características del Teorema, pero se pueden encontrar factores primos grandes.

Se modifica el Teorema General de La factorización de Cordero en la construcción de este software con el objetivo de evitar el uso de n fraccionarios o decimales. Se modifica la fórmula de n, se utiliza $n = \boldsymbol{s\big((r-1)s+r\big)x^2 + \big((r-1)s^2-(r-2)s-r\big)x + s\big((r-1)s+r\big)p-(r-1)s+[s^2x^2+s(s-2)x+ps^2-s+1]*r*T}$

Con $r, s, x, T \epsilon Z,\ s \neq 0,\ r \neq 0$ y $p \epsilon$ {2,3,5,11,17,41} . Y el número polinomial que se utiliza es $n^2 + (r-2)n + pr^2 - r + 1,\ \ r \neq 0\ y\ p\ \epsilon\ \{2, 3, 5, 11, 17, 41\}$.

5.4 PROGRAMA TEOREMA GENERAL AMPLIADO.

Su objetivo es factorizar números polinomiales de la forma

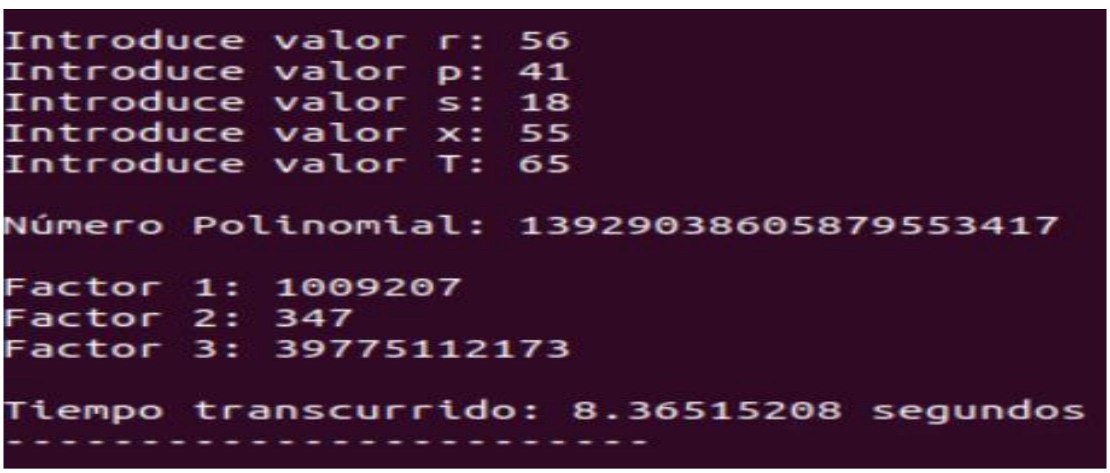
$n^2 + (r-2)n + pr^2 - r + 1,\ r \neq 0\ y\ p\ \epsilon\ \{2, 3, 5, 11, 17, 41\}$ utilizando la fórmula de

$n = s\big((r-1)s + r\big)x^2 + \big((r-1)s^2 - (r-2)s - r\big)x + s\big((r-1)s + r\big)p - (r-1)s +$ $[s^2x^2 + s(s-2)x + ps^2 - s + 1] * r * T$. Con $r, s, x, T\ \epsilon\ Z,\ s \neq 0,\ r \neq 0$ y $p\ \epsilon\ \{2, 3, 5, 11, 17, 41\}$

Por calculado el valor de n, se aplica el algoritmo r, n, p, para factorizar completamente el número polinomial. No utiliza las fórmulas para calcular $f_1\ y\ f_2$.

Su código se encuentra en el link:
https://github.com/roberto117343/R3C/blob/main/R3C_GENERAL_TEOREMA.cpp

Aplicación I.


```
Introduce valor r: 56
Introduce valor p: 41
Introduce valor s: 18
Introduce valor x: 55
Introduce valor T: 65

Número Polinomial: 13929038605879553417

Factor 1: 1009207
Factor 2: 347
Factor 3: 39775112173

Tiempo transcurrido: 8.36515208 segundos
-------------------------
```

El número polinomial factorizado tiene 20 dígitos. Y su tiempo de factorización es aproximadamente 8 segundos.

Aplicación II.

```
Introduce valor r: 48
Introduce valor p: 41
Introduce valor s: 23
Introduce valor x: 51
Introduce valor T: 60

Número Polinomial: 17354180272419717577

Factor 1: 1422229
Factor 2: 439
Factor 3: 73771
Factor 4: 53
Factor 5: 7109

Tiempo transcurrido: 10.12399575 segundos
-------------------------
```

El número polinomial tiene 20 dígitos.

Aplicación III

```
Introduce valor r: 46
Introduce valor p: 41
Introduce valor s: 20
Introduce valor x: 53
Introduce valor T: 57

Número Polinomial: 9572118118827334423

Factor 1: 19001
Factor 2: 9065325013
Factor 3: 911
Factor 4: 61

Tiempo transcurrido: 7.62483528 segundos
-------------------------
```

El número polinomial tiene 20 dígitos.

5.5 PROGRAMA MULTIPOLINOMIAL.

El software multi polinomial utiliza el algoritmo $r, n,\ p$ varias veces. La idea es que dado el valor de $p \epsilon \{2, 3, 5, 11, 17, 41\}$, (el usuario escoge un valor de p), el programa selecciona al azar varios valores de n y r, de acuerdo a la cantidad de iteraciones dadas. Construye varias veces los números polinomiales

$$r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1$$

$r \neq 0$ y $p \epsilon \{2, 3, 5, 11, 17, 41\}$ de acuerdo a la cantidad de iteraciones y los factoriza utilizando el Algoritmo de Cordero r, n, p.

Su código se localiza en el link:

https://github.com/roberto117343/R3C/blob/main/R3C_MULTIPOLINOMIAL.cpp

Es muy rápido en los cálculos, lo interesante es que el software obtiene los valores de r y n al azar, cuando se le da un valor inicial p.

Puede factorizar números polinomiales de más de 100 dígitos, y el número multi polinomial lo calcula multiplicando los factores de los números polinomiales que se obtienen.

Aplicación I

```
Introduce número de iteraciones: 3

Introduce el valor de p: 41

Número multipolinomial: 609735838997436183771841299119066234135993

Factores: 919 1399 2417 3067 5099 7333 5087741 336299621675041

Tiempo transcurrido: 0.10990875 segundos
------------------------
```

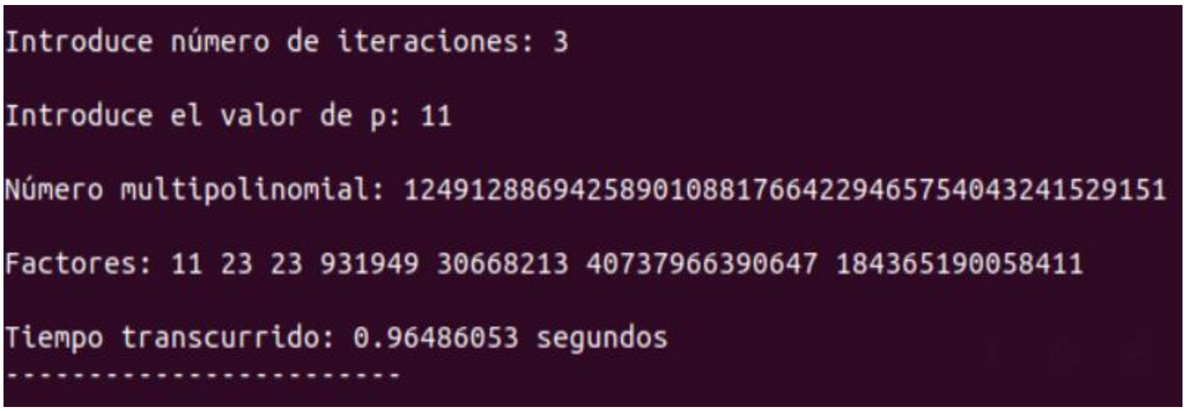
Para este caso el programa calcula secuencialmente 3 números polinomiales $r^2(n^2 + n + p) - (2n+1)r + 1$ $r \neq 0$ con $p = 41$ y los factoriza, el número multipolinomial es el resultado de multiplicar los factores obtenidos. El anterior número multi polinomial tiene 42 dígitos, por lo que aumenta considerablemente su cantidad de dígitos.

Aplicación II

```
Introduce número de iteraciones: 5

Introduce el valor de p: 17

Número multipolinomial: 222489108633455917370608114790383340689061273446114927172680726148622156667

Factores: 23 59 83 127 163 751 859 2371 87865597 16494655519 31447628837 1628031206699 84077945584907

Tiempo transcurrido: 0.85262254 segundos
------------------------
```

Para este caso el programa calcula secuencialmente 5 números polinomiales $r^2(n^2+n+p)-(2n+1)r+1\ \ r\neq 0$ con $p=17$ y los factoriza, el número multipolinomial es el resultado de multiplicar los factores obtenidos. El número multi polinomial tiene 74 dígitos, por lo que los números multi polinomiales aumentan considerablemente en su cantidad de dígitos. Su tiempo de cálculo es sumamente rápido, con menos de un segundo.

Aplicación III



```
Introduce número de iteraciones: 3

Introduce el valor de p: 11

Número multipolinomial: 1249128869425890108817664229465754043241529151

Factores: 11 23 23 931949 30668213 40737966390647 184365190058411

Tiempo transcurrido: 0.96486053 segundos
------------------------
```

El número multi polinomial tiene 46 dígitos, se utilizó $p=11$, y su tiempo de cálculo es de menos de un segundo, por lo que el software es eficiente en la factorización de números enteros.

6 EL CUARTO TEOREMA DE LA FACTORIZACIÓN DE CORDERO EN LOS NÚMEROS ENTEROS.

El Cuarto Teorema de La factorización de Cordero, es parte de la Publicación “Los Teoremas de La Factorización de Cordero en El Conjunto de los Números Enteros”, publicado en la Revista El Labrador de la Universidad Internacional San Isidro Labrador, https://revistaellabrador.net/index.php/RevistaElLabrador/article/view/57 Y publicado en la Generis Publishing como parte de un libro que se titula “Los Teoremas de la Factorización de Cordero en el Conjunto de los Números Enteros”. https://generis-publishing.com/upload/books/2022/05/LOS-TEOREMAS-DE-LA-FACTORIZACIN-DE-CORDERO-EN-EL-CONJUNTO-DE-LOS-NMEROS-ENTEROS_cover_1652294129.jpg

EL CUARTO TEOREMA DE LA FACTORIZACIÓN DE CORDERO EN EL CONJUNTOS DE LOS NÚMEROS ENTEROS

Sea $s, x, T \in \mathbb{Z}, s \neq 0 \ \ y \ p \in \{3,5,11,29\}$

1) Si $n = (4s^2x^2 + 4sx + 2ps^2 + 1) * T \pm \ (2sx^2 + x + sp)$
$entonces$: $2n^2 + p$ es compuesto y dos de sus factores tienen la forma:

$$f_1 = 4s^2x^2 + 4sx + 2ps^2 + 1$$

$$f_2 = \frac{2n^2 + p}{f_1} = 2f_1T^2 \pm 4\beta T + 2x^2 + p$$

$$\text{Con } \beta = 2sx^2 + x + sp$$

2) Si $n = (2x^2 + p) * T \pm \ (2sx^2 + x + sp)$
$entonces$: $2n^2 + p$ es compuesto y dos de sus factores tienen la forma:

$$f_1 = 2x^2 + p$$

$$f_2 = \frac{2n^2 + p}{f_1} = 2f_1T^2 \pm 4\beta T + 2f_1s^2 + 4sx + 1$$

$$\text{Con } \beta = 2sx^2 + x + sp$$

El Cuarto Teorema de La Factorización de Cordero permite factorizar números polinomiales de la forma $2n^2 + p$, donde $n \epsilon Z$ y $p \epsilon \{3, 5, 11, 29\}$. La factorización no es completa porque f_1 y f_2 pueden ser ambos, números primos, uno primo y el otro compuesto, o ambos compuestos. Para asegurar la factorización completa necesitamos los algoritmos de Cordero, el cuál aplicaremos más adelante.

Aplicación I

Parte 1)

Sea $s = 12, \ x = 8, \ p = 29, \ T = 5$

Calcular el valor de n

$$n = (4s^2x^2 + 4sx + 2ps^2 + 1) * T \pm \ (2sx^2 + x + sp)$$

$$n = (576x^2 + 48x + 8353) * 5 \pm (24x^2 + x + 348)$$

$$n = 45601 * 5 \pm 1892$$

Se pueden obtener dos valores para n.

$$n = 229897 \quad o \quad n = 226113$$

Se sabe por el Teorema que uno de los factores de los números polinomiales es 45601, el otro se puede encontrar por división o utilizando su fórmula correspondiente.

Los dos números polinomiales que se obtienen son:

$$2 * 229897^2 + 29 = 105705261247 = 45601 * 2318047$$

$$2 * 226113^2 + 29 = 102254177567 = 45601 * 2242367$$

Aplicación II

Parte 2)

Sea $x = 18,\ p = 11,\ T = 5, s = 4$

Calcular el valor de n

$$n = (2x^2 + p) * T \pm (2sx^2 + x + sp)$$

$$n = (2 * 18^2 + 11) * 5 \pm (8 * 18^2 + 18 + 44)$$

$$n = 659 * 5 \pm 2654$$

$$n = 5949 \ o \ n = 641$$

Se sabe por el Teorema que uno de los factores de los números polinomiales es 659, el otro se puede encontrar por división o utilizando su fórmula correspondiente.

Los dos números polinomiales que se obtienen son:

$$2 * 5949^2 + 11 = 70781213 = 659 * 107407$$

$$2 * 641^2 + 11 = 821773 = 659 * 1247$$

7 ALGORITMO Y SOFTWARE PARA EL CUARTO TEOREMA DE LA FACTORIZACIÓN DE CORDERO EN EL CONJUNTO DE LOS NÚMEROS ENTEROS.

El Algoritmo está basado en el Cuarto Teorema de la Factorización de Cordero, y a partir del algoritmo se construye el Software.

7.1 ALGORITMO PARA EL CUARTO TEOREMA DE LA FACTORIZACIÓN DE CORDERO.

Sea $n \epsilon Z$, $p \epsilon \{3,5,11,29\}$.

$$\emptyset = \sqrt{-32pT_{\emptyset}{}^2 + 16(|n| - 2p)T_{\emptyset} + 8(|n| - p) + 1}$$

I. Si para algún $T_{\emptyset} \epsilon Z,\ T_{\emptyset} \geq 0$ existe $\emptyset \epsilon N$ entonces el número polinomial $2n^2 + p$ es compuesto y hay que calcular $\frac{t}{b} = \frac{-1 \pm \emptyset}{4(2T_{\emptyset}+1)}$una fracción canónica, o t y b primos relativos donde $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor del número polinomial.

II. Si no existe algún $T_{\emptyset} \epsilon Z,\ T_{\emptyset} \geq 0$ tal que $\emptyset \epsilon N$ entonces utilizamos el segundo algoritmo.

$$\emptyset = \sqrt{-8p(2T_{\emptyset} + 1)^2 + 8(|n| - 2p)(2T_{\emptyset} + 1) + 8(|n| - p) + 1}$$

III. Si para algún $T_{\emptyset} \epsilon Z,\ T_{\emptyset} \geq 0$ existe $\emptyset \epsilon N$ entonces el número polinomial $2n^2 + p$ es compuesto y hay que calcular $\frac{t}{b} = \frac{-1 \pm \emptyset}{8(T_{\emptyset}+1)}$una fracción canónica, o t y b primos relativos donde $2t^2 + pb^2$ o $\frac{2t^2+pb^2}{2}$ es un factor del número polinomial.

Aplicación I

Sea $n = 5674 \quad y \quad p = 29$

Utilizando los algoritmos encontramos con el primer algoritmo los pares ordenados:

$$(85{,}989) \quad y \quad (91, 733)$$

El número polinomial es $2 * 5674^2 + 29 = 64388581$

Para $(85{,}989)$

$$\frac{t}{b} = \frac{-1 + \emptyset}{4(2T_{\emptyset} + 1)} = \frac{-1 + 989}{4 * (2 * 85 + 1)} = \frac{13}{9}$$

Uno de los factores primos es

$$2t^2 + pb^2 = 2 * 13^2 + 29 * 9^2 = 2687$$

$$\frac{t}{b} = \frac{-1 - \emptyset}{4(2T_{\emptyset} + 1)} = \frac{-1 - 989}{4 * (2 * 85 + 1)} = \frac{-55}{38}$$

Uno de los factores primos es

$$\frac{2t^2 + pb^2}{2} = \frac{2 * 55^2 + 29 * 38^2}{2} = 23963$$

Para $(91{,}733)$

$$\frac{t}{b} = \frac{-1 + \emptyset}{4(2T_{\emptyset} + 1)} = \frac{-1 + 733}{4 * (2 * 91 + 1)} = 1$$

Uno de los factores primos es

$$2t^2 + pb^2 = 2 * 1^2 + 29 = 31$$

$$\frac{t}{b} = \frac{-1 - \emptyset}{4(2T_{\emptyset} + 1)} = \frac{-1 - 733}{4 * (2 * 91 + 1)} = \frac{-367}{366}$$

Uno de los factores primos es

$$\frac{2t^2 + pb^2}{2} = \frac{2 * 367^2 + 29 * 366^2}{2} = 2077051$$

Entonces la factorización completa de:

$$2 * 5674^2 + 29 = 64388581 = 31 * 2687 * 773$$

Aplicación II

Sea $n = 2341 \quad y \quad p = 29$

Utilizando los algoritmos no encontramos pares ordenados, por lo que:

El número polinomial es $2 * 2341^2 + 29 = 10960591$ es un número primo.

Aplicación III

Sea $n = 1239 \quad y \quad p = 29$

Utilizando los algoritmos encontramos con el segundo algoritmo el único par ordenado:

$$(19,159)$$

El número polinomial es $2 * 1239^2 + 29 = 3070271$ es biprimo.

$$\frac{t}{b} = \frac{-1 + \emptyset}{8(T_{\emptyset} + 1)} = \frac{-1 + 159}{8 * (19 + 1)} = \frac{79}{80}$$

Uno de los factores primos es

$$\frac{2t^2 + pb^2}{2} = \frac{2 * 79^2 + 29 * 80^2}{2} = 99041$$

$$\frac{t}{b} = \frac{-1 - \emptyset}{8(2T_{\emptyset} + 1)} = \frac{-1 - 159}{8 * (19 + 1)} = -1$$

El otro factor primo es

$$2t^2 + pb^2 = 2 * 1^2 + 29 = 31$$

Entonces la factorización completa de:

$2 * 1239^2 + 29 = 3070271 = 31 * 99041$

7.2 SOFTWARE $\boldsymbol{n,p}$ PARA FACTORIZAR COMPLETAMENTE NÚMEROS POLINOMIALES DE LA FORMA $\boldsymbol{2n^2 + p}$ con $\boldsymbol{n \epsilon Z}$ y $\boldsymbol{p \epsilon \{3, 5, 11, 29\}}$.

El algoritmo, n, p es sorprendente por su sencillez. Solamente necesita el valor de p y el valor de n.

Se localiza en el link:

https://github.com/roberto117343/R3C/blob/main/R3C_CUARTO_TEOREMA.cpp

Aplicación I

Sea $n = 678765409\ y\ p = 29$

```
Introduce valor p: 29
Introduce valor n: 678765409

Número Polinomial: 921444960909874591

Factor 1: 30057439
Factor 2: 34171
Factor 3: 24247
Factor 4: 37

Tiempo transcurrido: 1.10612343 segundos
------------------------
```

El número polinomial tiene 18 dígitos y su tiempo de cálculo es un poco más de un segundo.

Aplicación II

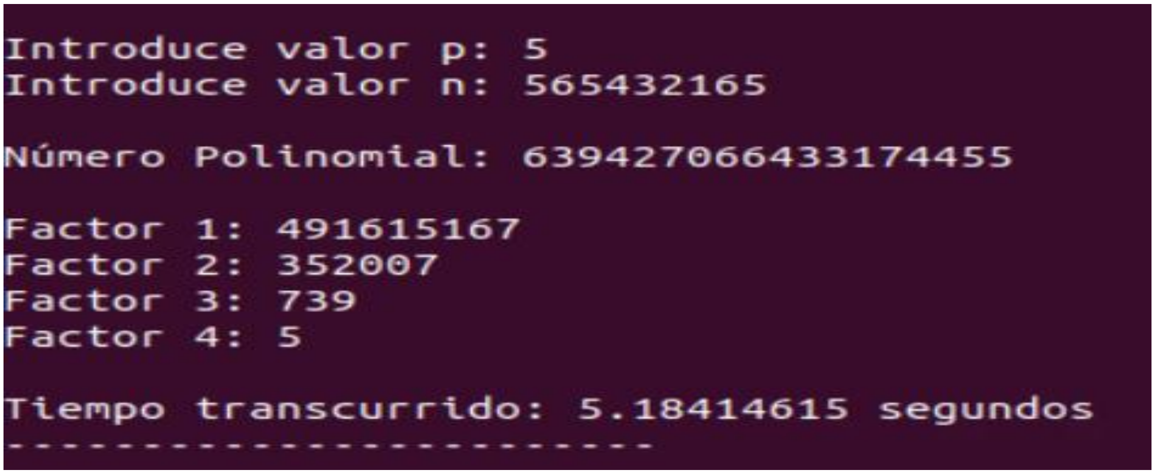
Sea $n = 456543219\ y\ p = 11$

```
Introduce valor p: 11
Introduce valor n: 456543219

Número Polinomial: 416863421629763933

Es biprimo

Factor 1: 11
Factor 2: 37896674693614903

Tiempo transcurrido: 1.78858632 segundos
------------------------
```

Número polinomial de 18 dígitos.

Aplicación III

Sea $n = 565432165\ y\ p = 5$



Número polinomial de 18 dígitos.

8. CONCLUSIONES.

1. Con la utilización del Teorema General de La factorización de Cordero, los Algoritmos de Cordero y los Softwares construidos por el Bioinformático Roberto Reinosa Fernández se logra comprobar la eficacia del Teorema y los Algoritmos en la factorización prima de los números polinomiales de la forma $r^2n^2 + r(r-2)n + pr^2 - r + 1 = r^2(n^2 + n + p) - (2n+1)r + 1$
2. La investigación abre un nuevo camino en el estudio de los polinomios generadores de números primos.
3. Los Teoremas, fórmulas y procedimientos Algorítmicos descubiertos por el matemático Ronald Cordero Méndez contribuyen a la Teoría de Números en el difícil problema de encontrar una solución definitiva a la factorización prima de los números enteros.
4. El Cuarto Teorema de la Factorización de Cordero en el Conjunto de los Números Enteros, sus Algoritmos y los Softwares construidos a partir de este Teorema colaboran en la factorización prima de los números polinomiales de la forma $2n^2 + p$.
5. La Investigación fomenta el interés por el estudio de los polinomios generadores de números primos, y su ya comprobada utilidad en la factorización de los números enteros.
6. Reconocemos que todavía falta mucho por caminar en el duro proceso de encontrar una solución definitiva en la factorización de los números enteros, pero la investigación contribuye a la búsqueda de tan esperada solución.

9. BIBLIOGRAFÍA.

- Abel, U. y Siebert, H.”Secuencias con un gran número de valores primos”. Soy. Matemáticas. Mensual 100, 167-169, 1993.
- Boston, N. y Greenwood, M. L. “Cuadráticas que representan números primos”. América. Matemáticas. Mensual 102,595-599, 1995
- Dudley, U. “Historia de la fórmula de los números primos”. América. Matemáticas. Mensual 76, 23-28, 1969.
- Garrison, B. “Polinomios con un gran número de valores primos”. América. Matemáticas. Mensual 97, 316-317, 1990.
- Hardy, G. H. y Wright, E. M. “Introducción a la Teoría de Números”, 5° ed. Oxford, Inglaterra: Clarendon Press, 1979.
- Pregg, E. Jr. “Concursos de programación de Al Zimmermann: polinomios generadores de primos”. 13 de marzo de 2006. https:// www.recmath.org/contest/description.php.