Universidad Nacional de Colombia
Departamento de Matemáticas
Álgebra Abstracta y Computacional
Taller Semana 3 (2022-II)

Prof. José L. Ramírez

. . . .

Justificar cada una de sus respuestas.

1. (Algoritmo de Euclides) Implemente dos funciones `AlgEuclides`$[a, b]$ y `AlgExtEuclides`$[a, b]$ en *Mathematica*® con el Algoritmo de Euclides y el Algoritmo de Euclides extendido de tal forma que `AlgEuclides` imprima como resultado los cocientes y residuos q_i y r_i. Mientras que `AlgExtEuclides` debe imprimir una tabla con los valores $i, x_i, y_i, ax_i + by_i$ (como el ejemplo visto en clase). Considere dos enteros

   ```
   In[1]:= a=Random[Integer, {10, 10^5}];

   In[2]:= b=Random[Integer, {10, 10^5}];
   ```

   Calcule el máximo común divisor de a y b con la función `AlgEuclides` y GCD (la que ya está implementada en *Mathematica*®). Compare los tiempos de ejecución. Repita esto con los siguientes enteros:

   ```
   In[3]:= a=Random[Integer, {10^99, 10^100}];

   In[4]:= b=Random[Integer, {10^99, 10^100}];
   ```

2. Ejercicio 2, pág. 145, Ref [1].

3. (Método Binario para calcular GCD) Un algoritmo alterno para calcular el máximo común divisor de dos enteros a y b se basa en la aplicación iterada del siguiente procedimiento. Sin perdida de generalidad vamos a suponer que $a, b > 0$.

   [1]

[1] Una de las ventajas de este algoritmo radica en que no se deben hacer cocientes, como en el caso del Algoritmo de Euclides.

Algoritmo 1 Algoritmo Binario para GCD

Input: $a, b \in \mathbb{Z}$

Output: $\mathsf{mcd}(a, b) \in \mathbb{Z}$

procedure EuclidesBinario(a, b)
    if $a = b$ **then return** a
    end if
    if $a \equiv 0 \pmod 2$ **and** $b \equiv 0 \pmod 2$ **then return** $2\mathsf{mcd}(a/2, b/2)$
    end if
    if exactamente uno de los dos números es par, digamos a, pero NO b **then return** $\mathsf{mcd}(a/2, b)$
    end if
    if $a \equiv 1 \pmod 2$ **and** $b \equiv 1 \pmod 2$, y supongamos que $a > b$ **then return** $\mathsf{mcd}((a-b)/2, b)$
    end if
end procedure

a) Utilice el anterior algoritmo para calcular (a mano!) el m.c.d de los números 8771 y 3206.

b) Justifique la razón (matemática) por la que el algoritmo es correcto.

c) Demuestre que la complejidad del algoritmo es $O(\max\{\mathsf{Lon}(a), \mathsf{Lon}(b)\})$.

d) () Sean x, y números enteros positivos. Escriba una función `GCDBinario[x,y]` en la que implemente este algoritmo (el algoritmo debe imprimir los pasos intermedios). Con la función diseñada calcule el m.c.d. de los números 8771 y 3206.

e) () Considere dos enteros

```
In[5]:= a=Random[Integer, {10, 10^5}];
In[6]:= b=Random[Integer, {10, 10^5}];
```

Calcule el máximo común divisor de a y b con la función `GCDBinario[x,y]` y *GCD* (la que ya está implementada en *Mathematica*®). Compare los tiempos de ejecución. Repita esto con los siguientes enteros:

```
In[7]:= a=Random[Integer, {10^99, 10^100}];
In[8]:= b=Random[Integer, {10^99, 10^100}];
```

4. Sean a y b enteros positivos con $a > b$.

a) Demuestre que $\texttt{Euc}(a, b)$ es a lo más 5 veces el tamaño de b (en base 10). () Verifique su resultado experimentalmente en *Mathematica*®.

b) Sean m y n enteros positivos. Escriba una función `VisualEuc`$[m, n]$ que grafique todas las parejas ordenadas (a, b), con $|a|, |b| \leq m$ tales que `Euc`$(a, b) = n$. Muestre las gráficas para $n = 1, 2, 3, 4, 5$ y $m = 1000$.

5. En el año 2016, Javier Cilleruelo y Florian Luca demostraron que todo entero positivo es suma de tres capicúas (palindromos) en cualquier base ≤ 5. Por ejemplo, en base 10 tenemos que 100 se puede escribir como suma de (77,22,1) (todos capicúas) (tenga en cuenta que (22,77,1) se considera una solución diferente). Calcule el total de formas de representar 800 como suma de tres capicúas mayores o iguales que cero. ¿Cuántas de estas soluciones están formadas únicamente por números primos? Muestre todas las soluciones para este último caso. Repita estas mismas preguntas para 2018.

   Si en este problema consideramos que $(22, 77, 1)$ es la misma solución que $(77, 22, 1)$ para $n = 100$. Calcule el total de formas de representar 800 como suma de tres capicúas mayores o iguales que cero. Repita esta misma pregunta para 2018.

. . . .

[Ref 1] K.O. Geddes, S. R. Czapor, G. Labahn. Algorithms for Computer Algebra. Kluwer Academic Publisher, 1992.

[Ref 2] E. A. Lamagna. Computer Algebra, Concepts and Techniques. CRC Press, 2019.