PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283
Interrogación 2

El plazo de entrega de la interrogación es el jueves 19 de noviembre a las 5pm. Para entregar la interrogación debe utilizar el mismo repositorio de git que usted usó para entregar la interrogación 1. La interrogación tiene 4 preguntas; las soluciones de estas preguntas deben ser colocadas en el directorio `I2` de su repositorio con los nombres `p1.ext`, `p2.ext`, `p3.ext` y `p4.ext`, donde `ext` es la extensión correspondiente al tipo de archivos que usted suba en su repositorio (se recomienda utilizar `pdf` o `jpeg`). Las respuestas pueden ser escritas a mano o en la Latex (no se recomienda el uso de Word, pero no está prohibido). Para realizar consultas sobre la interrogación utilice el foro `https://github.com/PUC-IIC2283/2020-IIC2283-Pruebas/issues`.

1. Dado un número natural $n \geq 2$, decimos que un a es un divisor de 0 en módulo n si y solo si $a \not\equiv 0 \bmod n$ y existe b tal que $b \not\equiv 0 \bmod n$ y $a \cdot b \equiv 0 \bmod n$. Por ejemplo, el número 2 es divisor del 0 en módulo 6 ya que $2 \not\equiv 0 \bmod 6$, $3 \not\equiv 0 \bmod 6$ y $2 \cdot 3 \equiv 0 \bmod 6$.

   (a) [0.6 puntos] Sea $n \geq 2$ un número natural. Demuestre que n es un número primo si y solo si n no tiene divisores del 0.

   (b) [0.6 puntos] Pregunta eliminada por un error en el enunciado. No es necesario que la responda, el puntaje le será asignado de manera automática.

   (c) [0.6 puntos] Sean p, q dos números primos distintos y $n = p \cdot q$. Demuestre que el polinomio $x^2 - 1$ tiene a los más 4 raíces en el conjunto $\{1, \ldots, n-1\}$.

2. Dada una lista $L = [a_1, \ldots, a_n]$ tal que $n \geq 1$ y cada $a_i > 0$, el máximo común divisor de L se define como el mayor número natural que divide a todos los números en L. Considere el siguiente algoritmo para calcular el máximo común divisor de una lista:

```
MCDLista(L)
    [a_1, ..., a_n] := L
    if n = 1 then return a_1
    else if n = 2 then return MCD(a_1, a_2)
    else
        k := ⌊n/2⌋
        b := MCDLista([a_1, ..., a_k])
        c := MCDLista([a_{k+1}, ..., a_n])
        return MCD(b, c)
```

Nótese que **MCDLista** utiliza al algoritmo **MCD** para calcular el máximo común divisor entre dos números, el cual fue estudiado en clases.

(a) [0.7 puntos] Demuestre que el algoritmo **MCDLista** es correcto.

(b) [0.7 puntos] Considerando el cálculo de la función $x \bmod y$ como la operación básica para el algoritmo **MCDLista**, encuentre una función $g(m)$ tal que el tiempo de ejecución del algoritmo es $O(g(m))$, donde m es el tamaño de la entrada. Además, justifique por qué la función $g(m)$ satisface esta condición.

3. [1 punto] Dado un grupo $(G, \circ)$, un elemento $a \in G$ y $n \in \mathbb{N}$, el término a^n es utilizado para denotar:

$$\underbrace{a \circ a \circ \cdots \circ a}_{n \text{ veces}}$$

En particular, a^0 se define como el neutro del grupo y $a^1 = a$. Por ejemplo, para $(\mathbb{Z}_8, +)$ tenemos que $3^0 = 0$, $3^1 = 3$, $3^2 = 6$ y $3^3 = 1$. Además, $(G, \circ)$ se dice cíclico si existe $g \in G$ tal que para todo $a \in G$, existe $k \in \mathbb{N}$ tal que $a = g^k$, y en este caso g se denomina un generador del grupo. Por ejemplo, 1 y 3 son generadores del grupo $(\mathbb{Z}_8, +)$, por lo que este grupo es cíclico.

Sea $n \geq 2$ un número natural y $a \in \mathbb{Z}_n$. Demuestre que a es un generador de $(\mathbb{Z}_n, +)$ si y solo si $\text{MCD}(a, n) = 1$.

4. Del ejercicio anterior se deduce que el número de generadores de $(\mathbb{Z}_n, +)$ es $\phi(n) = |\mathbb{Z}_n^*|$. Este resultado se puede extender a cualquier grupo finito $(G, \circ)$ que es cíclico: el número de generadores del grupo es $\phi(|G|)$. En particular, si p es un número primo, entonces $(\{1, \ldots, p-1\}, \cdot)$ es un grupo cíclico con $\phi(p-1)$ generadores.

Dado un número primo p, considere el siguiente algoritmo aleatorizado para construir un generador del grupo cíclico $(\{1, \ldots, p-1\}, \cdot)$:

Gen(p)
Escoja al azar y con distribución uniforme un elemento $a \in \{1, \ldots, p-1\}$
return a

Responda las siguientes preguntas sobre este algoritmo.

(a) [0.6 puntos] Suponiendo que tiene un procedimiento **LanzarMoneda**() que retorna 0 ó 1 con probabilidad $\frac{1}{2}$ y que funciona en tiempo $O(1)$, explique cómo se puede implementar el primer paso del algoritmo en tiempo $O(\log p)$.

(b) [0.6 puntos] **Gen** es un algoritmo de tipo Monte Carlo ya que **Gen**(p) puede retornar un número que no es un generador del grupo $(\{1, \ldots, p-1\}, \cdot)$. Encuentre una función $f(p)$ tal que:

$$\Pr(\textbf{Gen}(p) \text{ retorne un generador de } (\{1, \ldots, p-1\}, \cdot)) \quad \geq \quad f(p)$$

La funcion $f(p)$ no debe ser nula, vale decir, se debe cumplir que $f(p) > 0$ para todo $p \geq n_0$, donde n_0 es un número natural fijo.

(c) [0.6 puntos] Suponga que tiene un algoritmo **VerGen**(p, a) que, dado un número primo p y $a \in \{1, \ldots, p-1\}$, verifica is a es un generador del grupo $(\{1, \ldots, p-1\}, \cdot)$. Utilizando este algoritmo, se define el siguiente algoritmo aleatorizado para construir un generador del grupo $(\{1, \ldots, p-1\}, \cdot)$:

GenCorr(p)
    $a := 1$
    while VerGen(p, a) = **false do**
        Escoja al azar y con distribución uniforme un elemento $a \in \{1, \ldots, p-1\}$
    return a

Sea T_p una variable aleatoria que representa el número de veces que debe realizarse el ciclo **while** en la llamada **GenCorr**(p). Vale decir, $T_p = i$ si el ciclo se ejecuta i veces, donde $i \geq 0$. Demuestre que $\mathrm{E}(T_p) \in O(\log p)$.