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

Diseño y Análisis de Algoritmos - IIC2283
Examen

1. El siguiente es el algoritmo visto en clases para calcular la transformada rápida de Fourier:

FFT($\bar{a}$)
  $(a_0, \ldots, a_{n-1}) := \bar{a}$
  if $n = 2$ **then**
    $y_0 = a_0 + a_1$
    $y_1 = a_0 - a_1$
    return $[y_0, y_1]$
  else
    $\bar{a}_0 := (a_0, \ldots, a_{n-2})$
    $\bar{a}_1 := (a_1, \ldots, a_{n-1})$
    $[y_{0,0}, \ldots, y_{0,\frac{n}{2}-1}] :=$ **FFT**($\bar{a}_0$)
    $[y_{1,0}, \ldots, y_{1,\frac{n}{2}-1}] :=$ **FFT**($\bar{a}_1$)
    $\omega_n := e^{\frac{2\pi i}{n}}$
    $\alpha := 1$
    for $k := 0$ **to** $\frac{n}{2} - 1$ **do**
      $y_k := y_{0,k} + \alpha \cdot y_{1,k}$
      $y_{\frac{n}{2}+k} := y_{0,k} - \alpha \cdot y_{1,k}$
      $\alpha := \alpha \cdot \omega_n$
    return $[y_0, \ldots, y_{n-1}]$

(a) [0.5 puntos] Indique qué recibe como entrada y qué retorna **FFT**.

Corrección: Se asigna 0.5 puntos por la respuesta correcta a la pregunta

(b) [0.5 puntos] Explique los pasos del algoritmo **FFT**.

Corrección: Se asigna 0.5 puntos por explicar los pasos del algoritmo.

(c) [0.5 puntos] Explique por qué el algoritmo **FFT** es correcto. No es necesario que haga una demostración matemática aquí, sea breve e indique cuáles son las ideas centrales que muestran que el algoritmo es correcto.

Corrección: Se asigna 0.5 puntos por explicar por qué el algoritmo es correcto, en particular mencionando que n debe ser una potencia de 2, el algoritmo evalúa el polinomio

$p(x) = \sum_{i=0}^{n-1} a_i x^i$ en las n-raíces de la unidad, y el algoritmo realiza dos llamadas recursivas con tuplas de largo $\frac{n}{2}$ ya que las $\frac{n}{2}$-raíces de la unidad pueden ser obtenidas elevando al cuadrado las n-raíces de la unidad.

2. En esta pregunta usted va a analizar un algoritmo para estimar la probabilidad que una moneda cargada retorne cara. De manera más precisa, sea **Moneda**( ) un procedimiento que retorna 1 con probabilidad p y 0 con probabilidad $1 - p$, donde $p \in [0, 1]$, y defina **EstimarMoneda**( ) como el siguiente algoritmo:

   EstimarMoneda(n)
   $\quad sum := 0$
   $\quad$ **for** $i := 1$ **to** n **do**
   $\quad\quad sum := sum +$ **Moneda**( )
   $\quad$ **return** $\frac{sum}{n}$

   (a) [0.8 puntos] Dado $\varepsilon \in (0, 1)$, demuestre que:

   $$\Pr(|\textbf{EstimarMoneda}(n) - p| < \varepsilon) \quad \geq \quad 1 - \frac{p(1-p)}{n\varepsilon^2}.$$

   Corrección: El puntaje de esta pregunta se asigna de la siguiente forma:

   - [0.4 puntos] Se explica cómo utilizar la desigualdad de Chebyshev en este problema.
   - [0.8 puntos] Se explica cómo utilizar la desigualdad de Chebyshev en este problema, y se utiliza correctamente para demostrar la cota inferior.

   (b) [0.7 puntos] Calcule un valor de n tal que para todo $p \in [0, 1]$, el valor retornado por **EstimarMoneda**(n) tiene un error como estimación de p de a lo más $1\,\%$ con una probabilidad mayor o igual a $\frac{999}{1000}$. En símbolos, el valor de n encontrado debe satisfacer que:

   $$(\forall p \in [0, 1]) \ \Pr\left(|\textbf{EstimarMoneda}(n) - p| < \frac{1}{100}\right) \quad \geq \quad \frac{999}{1000}.$$

   Importante: para resolver esta pregunta puede usar (a) aunque no la haya resuelto.

   Corrección: El puntaje de esta pregunta se asigna de la siguiente forma:

   - [0.4 puntos] Se explica cómo utilizar (a) para obtener una cota inferior para el valor de n como una función de p.
   - [0.7 puntos] Se explica cómo utilizar (a) para obtener una cota inferior para el valor de n como una función de p, y se indica cómo seleccionar un valor de n que sea válido para todo $p \in [0, 1]$.

3. El siguiente es el algoritmo aleatorizado visto en clases para verificar si un número es primo:

   TestPrimalidad(n, k)
   $\quad$ **if** $n = 2$ **then return** PRIMO
   $\quad$ **else if** n es par **then return** COMPUESTO
   $\quad$ **else if** **EsPotencia**(n) **then return** COMPUESTO

```
else
    sea a_1, ..., a_k una secuencia de números elegidos de
            manera uniforme e independiente desde {1, ..., n − 1}
    for i := 1 to k do
        if MCD(a_i, n) > 1 then return COMPUESTO
        else b_i := EXP(a_i, (n−1)/2, n)
    neg := 0
    for i := 1 to k do
        if b_i ≡ −1 mod n then neg := neg + 1
        else if b_i ≢ 1 mod n then return COMPUESTO
    if neg = 0 then return COMPUESTO
    else return PRIMO
```

(a) [0.5 puntos] Indique qué recibe como entrada, cómo se puede equivocar y cuál es la probabilidad de error de **TestPrimalidad**.

Corrección: Se asigna 0.5 puntos por la respuesta correcta a la pregunta

(b) [0.5 puntos] Explique los pasos del algoritmo **TestPrimalidad**.

Corrección: Se asigna 0.5 puntos por explicar los pasos del algoritmo, y en particular indicar qué retornan los procedimientos auxiliares **EsPotencia**, **MCD** y **EXP**.

(c) [0.5 puntos] Explique por qué el algoritmo **TestPrimalidad** es correcto. No es necesario que haga una demostración matemática aquí, sea breve e indique cuáles son las ideas centrales que permiten acotar la probabilidad de error del algoritmo.

Corrección: Se asigna 0.5 puntos por explicar por qué el algoritmo es correcto, en particular definiendo los siguientes conjuntos:

$$\begin{aligned} S_n^+ &= \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv 1 \bmod n\}, \\ S_n^- &= \{a \in \mathbb{Z}_n^* \mid a^{\frac{n-1}{2}} \equiv -1 \bmod n\}, \end{aligned}$$

y mencionando cómo la probabilidad de error del algoritmo puede ser acotada considerando que si n es un número primo mayor o igual a 3, entonces $|S_n^+| = |S_n^-| = \frac{n-1}{2}$, y si n es un número compuesto que no es de la forma a^b con $b \geq 2$, entonces $|S_n^+ \cup S_n^-| \leq \frac{|\mathbb{Z}_n^*|}{2}$.

4. [1.5 punto] Demuestre que **Quicksort** en el caso promedio es $\Theta(n \cdot \log_2(n))$, considerando como la operación a contar la comparación y suponiendo que la entrada del algoritmo es una lista sin elementos repetidos.

Corrección: El puntaje de esta pregunta se asigna de la siguiente forma:

- [0.5 puntos] Se define lo que se debe demostrar en términos de la esperanza de una variable aleatoria.
- [1 punto] Se define lo que se debe demostrar en términos de la esperanza de una variable aleatoria, y se da la idea intuitiva de cómo hacer la demostración.

- [1.5 puntos] Se define lo que se debe demostrar en términos de la esperanza de una variable aleatoria, y se realiza la demostración completa.