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**.

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

   (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.

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

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

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

(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.

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

TestPrimalidad(n, k)
  if $n = 2$ **then return** PRIMO
  else if n es par **then return** COMPUESTO
  else if EsPotencia(n) **then return** COMPUESTO
  else
    sea $a_1, \ldots, a_k$ una secuencia de números elegidos de
        manera uniforme e independiente desde $\{1, \ldots, n-1\}$
    for $i := 1$ **to** k **do**
      if MCD(a_i, n) > 1 **then return** COMPUESTO
      else $b_i :=$ **EXP**($a_i, \frac{n-1}{2}, n$)
    $neg := 0$
    for $i := 1$ **to** k **do**
      if $b_i \equiv -1 \bmod n$ **then** $neg := neg + 1$
      else if $b_i \not\equiv 1 \bmod n$ **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**.
(b) [0.5 puntos] Explique los pasos del algoritmo **TestPrimalidad**.
(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.

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.