Este es el primer examen parcial del curso Programación Imperativa Modular (PIMO), 2018-2. El examen tiene un total de 4 preguntas. El examen es individual y no es permitido el uso de libros o apuntes.

Pregunta	Punto 1	Punto 2	Punto 3	Punto 4	Total
Puntos	10	15	15	15	55
Puntaje					

Nombre y Código: ____
Estimado: _______

Pregunta 1:

a. (2 puntos) Defina que significa que una función f pertenezca a O(f).
b. (3 puntos) Enuncie el Teorema Maestro.
c. (2 puntos) Clasifique ascendentemente por orden asintótico las siguientes funciones:

$$f_1 = n^3,\ f_2 = n^{0.3},\ f_3 = n,\ \ f_4 = \sqrt{n},\ \ f_5 = \frac{n^2}{\sqrt{n}},\ \ \ f_6 = n^2$$

d. (3 puntos) Use el Teorema Maestro para calcular T(n) en cada uno de los siguientes casos:
   - $T(n) = \ 3T\left(\frac{n}{2}\right) + \ n$
   - $T(n) = \ T\left(\frac{n}{2}\right) + \ O(1)$
   - $T(n) = \ 8T\left(\frac{n}{2}\right) + \ n^2$

Pregunta 2:

Considere la siguiente especificación:

- **Entrada:** Un arreglo $a[0..N)$ de números, N ≥ 0.
- **Salida:** El valor mínimo de $a[0..N)$, i.e., $(\downarrow i \mid 0 \ \leq i \ < N{:}\, a[i])$

El siguiente fragmento de código de Python implementa una solución del problema anterior:

```
def omin(a):
  N = len(a)
  r,n = float('inf'),0
  while n != N:
    r,n = min(r,a[n]),n+1
  return r
```

a. (5 puntos) En términos de notación O y justificando su respuesta, determine la complejidad temporal de la función omin.

b. (10 puntos) Considere los siguientes invariantes para el ciclo de las líneas 4-5

- $P_0: r = (\downarrow i \mid 0 \leq i < N: a[i])$
- $P_1: 0 \leq n \leq N$

Demuestre las propiedades de iniciación, estabilidad y terminación para P_0 y P_1. ¿Qué puede concluir al finalizar el ciclo?

Pregunta 3:

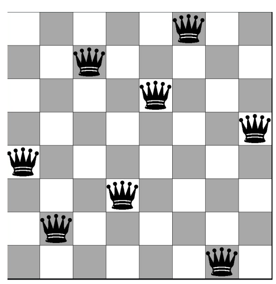
Considere la siguiente especificación:

- **Entrada:** Un arreglo $a[0..N)$ de números, N ≥ 0.
- **Salida:** Una permutación de los elementos originales en a, tales que $a'_1 \leq a'_2 \leq \ldots . \leq a'_n$

a. (10 puntos) Plantee un algoritmo con complejidad temporal $O(n \log n)$ que resuelva el problema especificado previamente. Debe demostrar la complejidad temporal del algoritmo.
b. (5 puntos) Modifique su algoritmo propuesto para adicionalmente contar la cantidad de inversiones que hay en el arreglo original.

Pregunta 4:

En un tablero de ajedrez de dimensiones de $n \ x \ n$, se desea colocar n reinas, de tal manera que no se amenacen entre ellas. Una solución para este problema en un tablero de 8 x 8 es:



a. (5 puntos) Encuentre una solución para este problema en un tablero de 4 x 4. Debe mostrar el árbol que se generó mientras buscaba la solución.
b. (10 puntos) Plantee un algoritmo que imprima todas las posibles soluciones para un tablero de $8 \ x \ 8$.