Universidad Autónoma de Yucatán
Facultad de Matemáticas
Maestría en Ciencias de la Computación
Examen de Matemáticas Computacionales

Nombre:____ Calificación: _________

<u>Instrucciones</u>: Lee cuidadosamente y responde correctamente lo que a continuación se te indica.

1. Determina el valor de verdad de las siguientes proposiciones. En caso de que el valor de verdad sea falso, escribe un contraejemplo que muestre la falsedad de la proposición.
   a. La relación R = {(1,3),(1,4),(2,3),(2,4),(3,1)} en el conjunto {1,2,3,4} es transitiva.
   b. Toda relación es una función.
   c. Toda función es una relación.
   d. La función $f: A \rightarrow P(A)$ definida por $f(x) = \{x\}$ es inyectiva. Nota: $P(A)$ es el conjunto potencia de A.

(20 puntos)

2. Prueba por inducción matemática que para n ∈ ℕ, se cumple que:

$$1^3 + 2^3 + 3^3 + \ldots + n^3 = \left(\frac{n(n+1)}{2}\right)^2$$

(15 puntos)

3. Construye una función $f: \mathbb{Z} \rightarrow \mathbb{Z}^+$ que sea inyectiva pero que no sea suprayectiva.

(15 puntos)

4. Considera la expansión de

$$\left(3x^2 - \frac{1}{x}\right)^9$$

   a. ¿Cuántos términos tiene la expansión del binomio?
   b. Encuentra el término constante en esta expansión.

(10 puntos)

5. ¿De cuántas maneras se pueden acomodar n hombres y n mujeres en una fila si los hombres y las mujeres deben estar alternados?

(10 puntos)

6. El algoritmo NumBinario encuentra la representación binaria de un entero decimal positivo, y se basa en divisiones sucesivas de 2. Así, para un entero positivo N, se calcula: $N=2q_0+r_0$, $q_0=2q_1+r_1$, $q_1=2q_2+r_2$, … $q_{k-1}=2q_k+r_k$, donde cada residuo r_i es 0 ó 1. El algoritmo se detiene cuando $q_k=0$. La representación binaria es entonces $N=r_k r_{k-1} \ldots r_1 r_0$.

   a) Determine el tamaño el problema y la operación básica. Justifique su elección.

   b) Obtenga la función de complejidad del algoritmo, de acuerdo a la operación básica seleccionada, y determine su orden.

```
procedimiento NumBinario(N,N2)
  inicio
        y ← N
        i ← 0
        mientras (y ≠ 0) haz
           si par(y) entonces
                r[i] ← 0
           sino
                r[i] ← 1
                y ← y - 1
           finsi
           y ← y/2
           i ← i + 1
        finm
        k ← i - 1
        N2 ← r[0…k]
fin.
```

(15 puntos)

7. El *selection sort* es un algoritmo de ordenamiento que localiza el elemento más pequeño de un arreglo de números y lo coloca en la primera posición, luego busca el siguiente elemento más pequeño y lo coloca en la segunda, y así sucesivamente. Considera el algoritmo del procedimiento para el *selection sort* y:
   a) describe cuál sería el mejor y el peor caso;
   b) determina la función de complejidad del algoritmo y su orden de complejidad para la operación básica de intercambio (swap).

```
procedure selection(a[ ],n);
begin                             /* a[0 … n-1] es el arreglo a ordenar */
    int i, j;
    int iMin;
   for (j = 0; j < n-1; j++)
       iMin = j;                  /* supone que el mínimo es el primer elemento */
      for ( i = j+1; i < n; i++)
          if (a[i] < a[iMin])     /* prueba contra los siguientes elementos para encontrar el menor */
             iMin = i;            /* encuentra el nuevo mínimo, recuerda su posición */
          endif
      endfor
      if (iMin != j)              /* Si no es el mismo elemento, intercambia los elementos*/
          swap(a[j], a[iMin]);
      endif
    endfor
  end
```

(15 puntos)