



**Lógica para Ciencia de la Computación - IIC2212**  
**Interrogación 3**

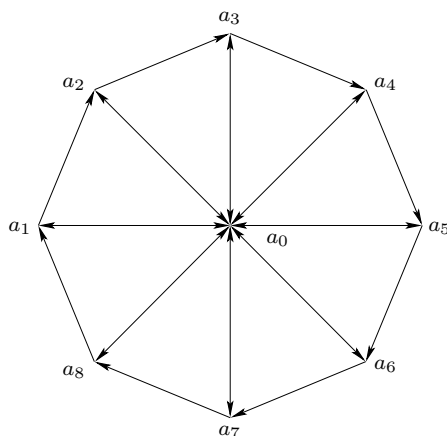
**Nota:** En esta prueba puede asumir como dados todos los problemas NP-completos que fueron mencionados en la guía, en el problema a resolver o en alguna clase o ayudantía.

**Problema 1 [1 punto]**

Un grafo no dirigido  $G = (A, N)$  es una  $n$ -rueda si se cumple que:

$$\begin{aligned} A &= \{a_0, a_1, \dots, a_n\}, \\ N &= \{(a_i, a_{i+1}) \mid i \in [1, n-1]\} \cup \{(a_n, a_1)\} \cup \{(a_0, a_i), (a_i, a_0) \mid i \in [1, n]\}. \end{aligned}$$

Por ejemplo, el siguiente grafo es una 8-rueda:



Dados grafos  $G_1 = (N_1, A_1)$  y  $G_2 = (N_2, A_2)$ , decimos que  $G_1$  es un subgrafo de  $G_2$  si  $N_1 \subseteq N_2$  y  $A_1 \subseteq A_2 \cap (N_1 \times N_1)$ . Defina el lenguaje SUB-RUEDA como

$$\text{SUB-RUEDA} = \{(G, n) \mid \text{existe un subgrafo } G' \text{ de } G \text{ tal que } G' \text{ es una } n\text{-rueda}\}.$$

Demuestre que SUB-RUEDA es NP-completo.

## Problema 2 [1 puntos]

Un grafo  $G_1 = (N_1, A_1)$  es isomorfo a un grafo  $G_2 = (N_2, A_2)$  si existe una biyección  $f : N_1 \rightarrow N_2$  tal que  $(a, b) \in A_1$  si y sólo si  $(f(a), f(b)) \in A_2$ . Defina el lenguaje SUB-ISO de la siguiente forma:

$$\text{SUB-ISO} = \{(G_1, G_2) \mid \text{existe un subgrafo } G'_1 \text{ de } G_1 \text{ tal que } G'_1 \text{ y } G_2 \text{ son isomorfos}\}.$$

¿Es SUB-ISO NP-completo?

## Problema 3 [2 puntos]

Dado un conjunto de números naturales  $S = \{a_1, \dots, a_n\}$  y un número natural  $b$ , defina la distancia entre  $S$  y  $b$  como:

$$d(S, b) = \min_{I \subseteq [1, n]} \left| \left( \sum_{i \in I} a_i \right) - b \right|.$$

Usando esta función de distancia, defina MAX-SUM como el siguiente problema de decisión:

$$\text{MAX-SUM} = \{(S, b, k) \mid d(S, b) \leq k\}.$$

Encuentre un problema de optimización asociado a MAX-SUM y demuestre que estos problemas son equivalentes. En particular, demuestre que poder solucionar MAX-SUM en tiempo polinomial permite encontrar eficientemente una solución que minimiza el valor de la función objetivo del problema de optimización.

## Problema 4 [2 puntos]

Dado un conjunto de números naturales  $S = \{a_1, \dots, a_n\}$  y un número natural  $b$ , decimos que  $S$  *representa*  $b$  si existe  $I \subseteq [1, n]$  tal que

$$\sum_{i \in I} a_i = b.$$

(a) [0.5 puntos] Dado  $b \geq 0$ , defina el lenguaje  $b$ -SUM como:

$$b\text{-SUM} = \{S \mid S \text{ representa } b\}.$$

¿Es  $b$ -SUM NP-completo?

(b) [1.5 puntos] Defina el lenguaje SUM como:

$$\text{SUM} = \{(S, b) \mid S \text{ representa } b\}.$$

Demuestre que SUM es NP-completo.

## Problema Adicional [2 puntos]

Recuerde que  $M_A$  es una MT universal que recibe como entrada  $C(M)0000w$ , simula  $M$  con entrada  $w$  y se detiene en un estado final si y sólo si  $M$  acepta  $w$ . Además, si  $M$  con entrada  $w$  retorna  $x$ , entonces  $M_A$  también retorna  $x$ . Usamos esta máquina para definir la complejidad de Kolmogorov de una palabra  $x \in \{0, 1\}^*$ :

$$K_A(x) = \min\{|y| \mid M_A \text{ con entrada } y \text{ retorna } x\},$$

Y usamos esta medida para introducir la noción de aleatoriedad. Formalmente, decimos que una palabra  $x$  es aleatoria si:

$$K_A(x) > \frac{|x|}{2}.$$

En clase demostramos que existen infinitas palabras aleatorias. Más aún, también demostramos que la mayor parte de las palabras son aleatorias, vale decir, la probabilidad de encontrar una palabra aleatoria de largo  $n$  entre todas las palabras de largo  $n$  tiende a 1 cuando  $n$  tiende a infinito.

Nótese que podemos obtener los mismos resultados si en lugar de considerar el alfabeto  $\{0, 1\}$  consideramos  $\{0, \dots, 9\}$ , vale decir, si usamos notación decimal en lugar de notación binaria. Usando todos estos resultados demuestre que existen infinitos números primos.