Presentación

Esta PEC profundiza en el concepto de complejidad computacional que cubre los contenidos estudiados en los módulos 6 y 7 de la asignatura. Los ejercicios trabajan los conceptos de medida de complejidad, la reducción y completitud, la clase NP-completo y algunos de los problemas intratables más importantes que se conocen.

Competencias

En esta PEC se trabajan las siguientes competencias del Grado de Ingeniería Informática:

- Capacidad para utilizar los fundamentos matemáticos, estadísticos y físicos para comprender los sistemas TIC.
- Capacidad para analizar un problema en el nivel de abstracción adecuado en cada situación y aplicar las habilidades y conocimientos adquiridos para resolverlo.

Objetivos

Los objetivos concretos de esta PEC son:

- Entender los conceptos de intratabilidad y no-determinismo.
- Conocer las diferentes clases de complejidad y saber clasificar los problemas en cada una de estas.
- Entender el concepto de reducción entre problemas y saber demostrar cuando un problema es NP-completo.
- Reconocer problemas intratables que aparecen de forma habitual en informática y en ingeniería.
- Entender y saber aplicar las técnicas básicas de reducción polinómica de los problemas NP-completos.

Descripción de la PEC a realizar

1. (Valoración de un 25 % = (2 %+2 %+3 %+8 %+10 %)

   Definimos el problema X como, dado un grafo $G = (V, A)$ y dos vértices $u, v \in V$, determinar la distancia entre u y v.

   Definimos el problema Y como, dado un grafo $G = (V, A)$, dos vértices $u, v \in V$, y un número natural k, determinar si hay un camino en G con como mucho k aristas que conecte u y v.

   a) Indica si X es un problema de decisión, de optimización, o ninguna de las dos cosas. Si es un problema de optimización, di también cuál es la función de valoración, y si se minimiza o maximiza.

   b) Indica si Y es un problema de decisión, de optimización, o ninguna de las dos cosas. Si es un problema de optimización, di también cuál es la función de valoración, y si se minimiza o maximiza.

   c) ¿Qué algoritmo de los vistos en los módulos usarías para resolver el problema X lo más eficientemente posible? En notación O, ¿qué coste tiene en términos de $|V|$ y $|A|$?

   d) Supongamos que tenemos un procedimento P_X que resuelve X: dado un grafo $G = (V, A)$ y dos vértices $u, v \in V$, devuelve $P_X(G, u, v)$, la distancia entre u y v.
   Usando P_X, implementa en seudo-código un procedimiento P_Y que resuelva Y:

   Entrada: grafo $G = (V, A)$; vértices $u, v \in V$; número natural k
   Salida: SÍ si hay un camino en G con como mucho k aristas que conecta u y v, NO en caso contrario.

   algoritmo $P_Y(G, u, v, k)$
       inicio
           ... (A COMPLETAR)
       fin

   Se valorará que uses el menor número posible de llamadas a P_X por llamada a P_Y. En el peor de los casos, ¿cuántas se hacen? Usa notación O.

   e) Supongamos que tenemos un procedimento Q_Y que resuelve Y: dado un grafo $G = (V, A)$, dos vértices $u, v \in V$, y un número natural k, devuelve $Q_Y(G, u, v, k)$, que es:

   - SÍ si hay un camino en G con como mucho k aristas que conecta u y v.
   - NO en caso contrario.

Usando Q_Y, implementa en seudo-código un procedimiento Q_X que resuelva X:

Entrada: grafo $G = (V, A)$; vértices $u, v \in V$
Salida: la distancia entre u y v

algoritmo $Q_X(G, u, v)$
  inicio
    ... (A COMPLETAR)
  fin

Se valorará que uses el menor número posible de llamadas a Q_Y por llamada a Q_X. En el peor de los casos, ¿cuántas se hacen? Usa notación O.

Solución:

a) Se trata de un problema de optimización. Se quiere minimizar la función de valoración siguiente: la longitud de los caminos que unen u y v.

b) Se trata de un problema de decisión, puesto que la respuesta es SÍ/NO.

c) El algoritmo BFS, con coste $O(|V| + |A|)$.

d) Una posible solución:

algoritmo $P_Y(G, u, v, k)$
  inicio
    retorno $P_X(G, u, v) \leq k$
  fin

Se hacen $O(1)$ llamadas a P_X por llamada a P_Y.

e) Una posible solución:

algoritmo $Q_X(G, u, v)$
  inicio
    si $\neg Q_Y(G, u, v, |V| - 1)$
      retorno ∞
    finsi
    $l \leftarrow 0$

$h \leftarrow |V| - 1$
mientras $l < h$
  $m \leftarrow (l + h)/2$
  si $Q_Y(G, u, v, m)$
    entonces $h \leftarrow m$
    sino $l \leftarrow m + 1$
  finsi
finmientras
retorno l
fin

Se hacen $O(\log(|V|))$ llamadas, puesto que en cada iteración la longitud del intervalo $[l, h]$ se reduce a la mitad.

---

2. (Valoración de un 25 % = (3 % + 5 % + 6 % + 8 % + 3 %))

El problema T se define así: dados $N > 0$, un vector de naturales $w = (w_0, ..., w_{N-1})$, un vector de naturales $v = (v_0, ..., v_{N-1})$, y dos naturales W y V, determinar si existe un subconjunto $S \subseteq \{0, ..., N-1\}$ tal que $\sum_{i \in S} w_i \leq W$ y $\sum_{i \in S} v_i \geq V$.

a) Identifica T con uno de los problemas vistos en los módulos. Describe del modo más preciso posible su clase de complejidad.

b) Completa el pseudo-código del siguiente procedimiento P_T para resolver T:

Entrada: vector de naturales $w = (w_0, ..., w_{N-1})$; vector de naturales $v = (v_0, ..., v_{N-1})$; naturales W y V
Salida: SÍ si existe un subconjunto $S \subseteq \{0, ..., N-1\}$ tal que $\sum_{i \in S} w_i \leq W$ y $\sum_{i \in S} v_i \geq V$, NO en caso contrario.

algoritmo $P_T(w, v, W, V)$
  inicio
    para $i \leftarrow 0$ **hasta** N
      para $j \leftarrow 0$ **hasta** W
        $m[i][j] = 0;$
      finpara
    finpara
    para $i \leftarrow 1$ **hasta** N
      para $j \leftarrow 0$ **hasta** W
        si $w[i-1] > j$
          entonces $m[i][j] \leftarrow m[i-1][j];$

sino $m[i][j] \leftarrow \text{máx}(\ m[i-1][j-w[i-1]] + v[i-1],\ \ m[i-1][j])$;
finsi
finpara
finpara
retorno ... (A COMPLETAR)
fin

Pista: la parte que falta se evalúa en tiempo $O(1)$.

c) El estudiante A afirma que el problema T es NP-Difícil. El estudiante B afirma que el procedimiento P_T tiene coste $O(N \cdot W)$. El estudiante C dice que A y B no pueden tener los dos la razón, porque eso implicaría que P = NP. Para cada estudiante di si tiene razón o no, y por qué.

d) El problema U se define así: dado un multiconjunto de $P > 0$ naturales $M = \{m_0, ..., m_{P-1}\}$, y un natural Q, determinar si existe un subconjunto S de M tal que la suma de todos los elementos de S sea igual a Q. Da una reducción polinómica de U a T indicando claramente qué entrada de T devuelve la reducción si se le pasa como argumento una entrada (M, Q) de U. Justifica por qué se trata de una reducción polinómica.

e) ¿Existe una reducción (no hace falta darla explícitamente) de T a U? ¿Por qué?

Solución:

a) T es el problema de KNAPSACK. Se trata de un problema NP-Completo.

b) Una posible solución: **retorno** $m[N][W] \geq V$;

c) El estudiante A tiene razón: al ser T NP-Completo, en particular es NP-Difícil.

También el estudiante B tiene razón. Como la pista indica que el código que falta tarda tiempo $O(1)$, el coste viene determinado por los dos bucles externos. Ambdos tienen el mismo coste asintótico: para una i y una j fijadas, el coste del código más interno es $O(1)$. Como el bucle interno itera $W+1$ veces y el externo $N+1$ veces, el coste es en efecto $O((N+1) \cdot (W+1)) = O(N \cdot W)$.

Por último el estudiante C no tiene razón. Es cierto que si T se pudiera resolver en tiempo polinómico entonces tendríamos que P = NP. Pero resulta que el algoritmo dado no resuelve T en tiempo polinómico, porque el valor de W es exponencial en el tamaño de W, o sea, el número de bits necesarios para representarlo.

d) Dada una entrada $(\{m_0, ..., m_{P-1}\}, Q)$ de U, la reducción devuelve la siguiente entrada de T: $((m_0, ..., m_{P-1}), (m_0, ..., m_{P-1}), Q, Q)$. Está claro que la reducción, que sólo duplica datos, tiene coste polinómico. Además, hay un subconjunto S de $\{0, ..., P-1\}$ tal que $\sum_{i \in S} m_i = Q$ si y sólo si hay un subconjunto S de $\{0, ..., P-1\}$ tal que $\sum_{i \in S} m_i \leq Q$ y $\sum_{i \in S} m_i \geq Q$. De modo que la respuesta de $(\{m_0, ..., m_{P-1}\}, Q)$ para U es SÍ si y sólo si la respuesta de $((m_0, ..., m_{P-1}), (m_0, ..., m_{P-1}), Q, Q)$ para T es SÍ. Se cumplen pues todas las propiedades de una reducción polinómica.

e) Sí, existe una reducción de T a U. Como se ha dicho antes, T es el problema de KNAPSACK, y U es el problema de SUMA-SUB. Se sabe que ambos son NP-Completos. En particular, como T pertenece a NP y U es NP-Difícil, la reducción que se nos pide debe existir.

---

3. (Valoración de un 25 % = (2 %+2 %+2 %+2 %+2 %)+(2,5 %+2,5 %)+(5 %+5 %))

   a) Para cada una de las siguientes fórmulas proposicionales, indica si está en **forma normal conjuntiva (FNC)**. Si lo está, indica el **número de cláusulas** y el **número de literales** de cada una. Si no lo está, explica brevemente por qué y cómo podría transformarse para que lo estuviera.

      1)
      $$F_1 = (a \vee \neg b) \ \wedge \ (b \vee c \vee d) \ \wedge \ (\neg a \vee \neg c)$$
      2)
      $$F_2 = (a \wedge b) \ \vee \ (\neg a \wedge c)$$
      3)
      $$F_3 = (\neg a \vee b) \ \wedge \ \neg(\neg b \vee c)$$
      4)
      $$F_4 = (a \leftrightarrow b) \ \vee \ (\neg b \wedge c)$$
      5)
      $$F_5 = (a \vee b) \ \wedge \ (c) \ \wedge \ (\neg d \vee \neg c \vee b)$$

   b) Sea la siguiente fórmula booleana en **Forma Normal Conjuntiva (FNC)**:

      $$F = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee x_4)$$

      1) Transforma F en una fórmula F' equivalente en **Forma Normal Disyuntiva (FND)**. Se valorará que la FND tenga el mínimo número de términos.
      2) Determina si F es satisfacible. En caso afirmativo, proporciona una **asignación de verdad** para las variables que satisfaga la fórmula.

c) Se consideran dos problemas numéricos clásicos:
- **Factorización:** dado un entero $N > 1$, encontrar (si existen) dos enteros $p, q > 1$ tales que $N = p \cdot q$.
- **Primalidad:** dado un entero $N > 1$, determinar si N es primo.

1) ¿Qué tipo de problema son? Indica si cada uno es un **problema de decisión** o de **optimización/construcción** y justifica tu respuesta.
2) ¿En términos de complejidad, en qué se diferencian?

Solución:

a) 1) F_1: Está en **FNC**, ya que es una conjunción de disyunciones de literales. Cláusulas:

$$C_1 = (a \vee \neg b), \quad C_2 = (b \vee c \vee d), \quad C_3 = (\neg a \vee \neg c)$$

Número de literales por cláusula: 2, 3 y 2 respectivamente.

2) F_2: La fórmula no está en Forma Normal Conjuntiva, ya que es una disyunción de conjunciones. Aplicando las leyes de distributividad, se obtiene una fórmula equivalente en FNC:

$$(a \vee \neg a) \wedge (a \vee c) \wedge (b \vee \neg a) \wedge (b \vee c)$$

Eliminando la cláusula tautológica $(a \vee \neg a)$, resulta la siguiente fórmula en Forma Normal Conjuntiva:

$$(a \vee c) \wedge (b \vee \neg a) \wedge (b \vee c)$$

3) F_3: No está en FNC. La negación $\neg(\neg b \vee c)$ no está en forma normal negativa; aplicando De Morgan:

$$\neg(\neg b \vee c) \equiv (b \wedge \neg c)$$

Entonces $F_3 \equiv (\neg a \vee b) \wedge b \wedge \neg c$, que podría dejarse en FNC como tres cláusulas: $(\neg a \vee b)$, (b) y $(\neg c)$.

4) F_4: No está en FNC, ya que contiene el bicondicional $\leftrightarrow$. Para llevarla a FNC habría que eliminarlo:

$$(a \leftrightarrow b) \equiv (\neg a \vee b) \wedge (\neg b \vee a)$$

y después distribuir la disyunción superior con $(\neg b \wedge c)$. Por tanto, la **FNC final simplificada** es:

$$(a \vee \neg b) \ \wedge \ (\neg a \vee b \vee c)$$

5) F_5: Está en **FNC**, con tres cláusulas:

$$C_1 = (a \vee b), \quad C_2 = (c), \quad C_3 = (\neg d \vee \neg c \vee b)$$

Número de literales: 2, 1 y 3 respectivamente.

b) 1) Dada la fórmula en FNC:

$$F = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee \neg x_3 \vee x_4),$$

queremos obtener una FND equivalente aplicando las leyes distributivas.
Denotamos las cláusulas como:

$$C_1 = (x_1 \vee \neg x_2 \vee x_3), \qquad C_2 = (\neg x_1 \vee x_2 \vee x_4), \qquad C_3 = (x_2 \vee \neg x_3 \vee x_4).$$

Paso 1: Distributiva entre C_1 y C_2

Aplicamos:

$$(A \vee B \vee C) \wedge (D \vee E \vee F) = \bigvee_{a \in \{A,B,C\},\ b \in \{D,E,F\}} (a \wedge b).$$

De este modo:

$$\begin{aligned} C_1 \wedge C_2 = {} & (x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (\neg x_2 \wedge \neg x_1) \vee \\ & (\neg x_2 \wedge x_4) \vee (x_3 \wedge \neg x_1) \vee (x_3 \wedge x_2) \vee (x_3 \wedge x_4), \end{aligned}$$

donde ya hemos eliminado los términos contradictorios $x_1 \wedge \neg x_1$ y $\neg x_2 \wedge x_2$.
Denotamos estos términos como $T_1, \ldots, T_7$.

Paso 2: Introducción de la tercera cláusula C_3

Ahora aplicamos la distributiva:

$$F = (C_1 \wedge C_2) \wedge C_3 = \bigvee_{i=1}^{7} \big(T_i \wedge (x_2 \vee \neg x_3 \vee x_4)\big).$$

Cada uno de estos términos se distribuye como:

$$T_i \wedge (x_2 \vee \neg x_3 \vee x_4) = (T_i \wedge x_2) \vee (T_i \wedge \neg x_3) \vee (T_i \wedge x_4).$$

A continuación se eliminan:
1. Conjunciones contradictorias $(x \wedge \neg x)$. 2. Términos absorbidos por otros más simples (por ejemplo, $x_1 \wedge x_2$ absorbe $x_1 \wedge x_2 \wedge \neg x_3$ y similares).
Tras simplificar todos los términos, obtenemos la FND mínima equivalente:

$$\boxed{F_{\text{FND}} \equiv (x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_3) \vee (x_3 \wedge x_4) \vee (x_4 \wedge \neg x_2) \vee (\neg x_1 \wedge \neg x_2 \wedge \neg x_3)}$$

2) La fórmula F es **satisfacible**. Una posible asignación de verdad que la satisface es:

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 1$$

Sustituyendo estos valores, cada cláusula de F se evalúa como verdadera, por lo que $F = 1$.

c) 1)
   - **Factorización:** es un problema de **construcción**, ya que se pide **encontrar** los factores de N. La salida no es booleana, sino una solución concreta (p, q).
   - **Primalidad:** es un problema de **decisión**, porque únicamente hay que responder si N es primo (sí/no). No requiere calcular factores, solo determinar su existencia.

   2) La **primalidad** es un problema **decisional** y actualmente **tractable**, ya que puede resolverse en tiempo polinómico; pertenece a la clase **P**. La **factorización** es un problema **constructivo** para el que no se conoce ningún algoritmo polinómico; su versión decisional (el problema FACTORES del módulo 7) pertenece a la clase **NP**. Esta diferencia tiene una consecuencia directa: podemos comprobar rápidamente si un número es primo, pero no podemos obtener fácilmente los factores de un número compuesto.

---

4. (Valoración de un 25 %) Parte online de la evaluación

En el Campus Virtual, dentro de la sección de “Contenidos” del aula de la asignatura, encontraréis la sección “PEC3 - Parte online”. Ahí hay un cuestionario con diversas preguntas que debéis resolver como último ejercicio de esta PEC.

Leed atentamente las siguientes instrucciones **antes de abrir el cuestionario**:

- Los contenidos que se evalúan en este cuestionario corresponden a los módulos 6 y 7 de la asignatura. Es importante que hayáis asimilado estos conocimientos **antes de abrir el cuestionario**.
- Para visualizar correctamente el cuestionario, utilizad Chrome o Firefox.
- El cuestionario estará abierto durante el plazo de la PEC y lo podéis resolver cuando queráis. De todas formas, una vez lo abráis tendréis un **tiempo limitado** para resolverlo (40 minutos).

  Importante: El cuestionario quedará cerrado a las 23:59 de la fecha límite de entrega. Si empezáis a hacerlo después de las 23:19 del último día, ¡tendréis menos de cuarenta minutos para hacerlo!
- Las respuestas a las preguntas se tienen que introducir directamente en el cuestionario. No es necesario que las entreguéis junto con el resto de respuestas de la PEC.
- Las preguntas del cuestionario son aleatorias: cada estudiante recibirá un enunciado diferente.
- En algunas preguntas tendréis que introducir la respuesta en un formato específico (p. ej. con los valores ordenados de una determinada forma y sin espacios). Es muy importante **seguir fielmente el formato indicado** a la hora de introducir vuestra respuesta.

- Disponéis de **2 intentos** para resolver el cuestionario. El objetivo de tener dos intentos es poder solventar posibles problemas que hayáis tenido en la realización del cuestionario, ya sean problemas técnicos o bien que hayáis abierto el cuestionario por error. Por tanto, debéis tener en cuenta que:
  - La nota que obtendréis en el cuestionario será la de vuestro último intento.
  - Después del 1r intento, no recibiréis la calificación obtenida ni recibiréis feedback sobre vuestra propuesta de solución. Por lo tanto, no recomendamos usar el 2o intento para intentar mejorar nota, ya que puede ser que obtengáis una nota inferior.
  - Si usáis el 2o intento, el enunciado que encontraréis será diferente del del 1r intento.
  - Podéis realizar los dos intentos en días diferentes, siempre que sea dentro del plazo de la PEC. Dispondréis de 40 minutos para cada intento.
  - Cada vez que iniciéis el cuestionario contará como un intento, aunque no enviéis la respuesta. Por ejemplo, **si habéis hecho el 1r intento y volvéis a abrir el cuestionario, invalidaréis vuestro 1r intento y os quedaréis con la nota del 2o**.

Recursos

Recursos Básicos

- Módulo didáctico 6. Complejidad computacional.
- Módulo didáctico 7. Problemas intratables.
- Colección de problemas

Recursos Complementarios

- PACs de semestres anteriores.

Criterios de valoración

- La PEC se tiene que resolver **de forma individual**. En caso que hayáis consultado recursos externos, es necesario referenciarlos.
- Es necesario justificar la respuesta de cada apartado. Se valorará tanto el resultado final como la justificación dada.
- En los apartados donde sea necesario aplicar algún algoritmo, se valorará la elección del algoritmo apropiado, los pasos intermedios, el resultado final y las conclusiones que se deriven. Hay que seguir el formato mostrado en los apuntes.
- Se considerará el uso de herramientas de inteligencia artificial (ChatGPT, etc.) como un acto de plagio.

Formato y fecha de entrega

Hay que entregar **un único documento** PDF con las respuestas de todos los ejercicios. El nombre del fichero tiene que ser: **PEC3_Apellido1Apellido2Nombre.pdf**. El documento no puede exceder las 10 páginas y no debe contener los enunciados, sólo la información necesaria para poder identificar a qué ejercicio se corresponde cada respuesta.

Este documento se tiene que entregar en la subsección **Entrega de la actividad PEC3** del "Reto 3. Los límites de la computación" **antes de las 23:59 del día 22/12/2025**. **Verificad que la entrega se ha realizado con éxito y que el documento entregado es el correcto. No se aceptarán entregas fuera de plazo.**