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 20 %=5 %+5 %+5 %+5 %)
   Clasificad los siguientes problemas decisionales según los siguientes tipos de complejidad temporal: Constante, Lineal, Cuadrática, Cúbica y Exponencial.

   a) Dadas dos cadenas de carácteres, decidir si son palíndromas.

   b) Dado un vector de enteros, decidir si existe un elemento en el vector tal que es igual a la suma de los elementos restantes.

   c) Dado un vector de enteros, decidir si existe un subconjunto no vacío de N elementos, tal que las suma de los elementos del subconjunto es positiva.

   d) Dados un vector de pesos p y otro de valores v con el mismo tamaño N, decidir si existe un subconjunto S no vacío de índices $i = 1, \ldots, N$, tal que la suma de los valores $\sum_{i \in S} v_i \geq k$ y la suma de los pesos $\sum_{i \in S} p_i \leq M$, y k, M son dos valores cualquiera.

Solución:

a) Lineal. Hay que comprobar todas la posiciones de la cadena de carácteres.

b) Lineal. Primero hay que sumar todos los elementos del vector y luego comprobar si la suma total menos el valor i-ésimo son iguales.

c) Lineal. Sólo hay que comprobar si existe un elemento positivo.

d) Es un problema NP-Completo. Concretamente el problema de la Mochila. La complejidad de los problemas NP-Completos es Exponencial.

---

2. (Valoración de un 20 %)

   Sean los siguientes problemas decisionales:

   a) $MULT2 = \{x \mid x \ mod \ 2 = 0\}$

   b) $MULT3 = \{x \mid x \ mod \ 3 = 0\}$

   c) $MOCHILA$

   d) $A = \{x \mid x \ mod \ 2 \neq 0\}$

   e) $B = \{MULT2 \cup A\}$

Justificad brevemente cuál de las siguientes reducciones son posibles:

a) MULT2 $\leq_p$MOCHILA

b) MULT2 $\leq_p$MULT3

c) MOCHILA $\leq_p$MULT3

d) MOCHILA $\leq_p$MOCHILA

e) B $\leq_p$A

f) MOCHILA $\leq_p$B

Solución: Primero vamos a clasificar los diferentes problemas:

a) MULT2 es el conjunto de los números múltiples de 2 y es P.

b) MULT3 es el conjunto de los números múltiples de 3 y es P.

c) MOCHILA es *NP-Completo*.

d) A es el conjunto de los impares y es P.

e) B es el conjunto vacío y es P.

Teniendo en cuenta lo anterior las respuestas son las siguientes:

a) Siempre es posible. Todo problema en P puede reducirse a uno NP mediante una $\leq_p$. La función de reducción, puede resolver el problema MULT2 con un tiempo polinómico ya que MULT2 pertenece a P. Si el resultado es cierto entonces devolvemos un resultado válido del problema NP, si es falso devolvemos un resultado inválido.

b) Siempre es posible. Una reducción de un problema P a otro en P siempre es posible mediante una $\leq_p$reducción.

c) Sólo sería posible si $P = NP$.

d) Sí que es posible. Todo problema puede reducirse a si mismo ya que la función identidad tiene un coste polinómico.

e) Sí que es posible. La reducción $\emptyset \leq_p A$ podría ser $f(x) = 2$.

f) No es posible. No existe una función de reducción de S $\leq_p \emptyset$ y $S \neq \emptyset$, y MOCHILA $\neq \emptyset$.

---

3. (Valoración de un 20 %=5 %+5 %+5 %+5 %)
Decid si son ciertas o falsas las siguientes afirmaciones y justificad brevemente las respuestas:

a) Si un problema es P entonces podemos asegurar que la complejidad espacial es polinómica.
b) Si un problema es $\mathsf{NP} - \mathsf{Completo}$ entonces podemos asegurar que la complejidad espacial es exponencial.
c) Si un problema es NP entonces seguro que lo podemos resolver con un tiempo polinómico.
d) Si un problema es NP entonces seguro que lo podemos resolver con un tiempo exponencial.

Solución:

a) Cierto ya que $P \subseteq PSPACE$.
b) Cierto ya que $NP \subseteq PSPACE$ y $PSPACE \subset EXPSPACE$.
c) Falso ya no sabemos si $P \subset NP$ o $P = NP$.
d) Cierto ya que $P \subset EXP$.

---

4. (Valoración de un 30 %=7.5 %+15 %+7.5 %)
Sean los siguientes problemas:

- A=*Dado un conjunto de N enteros y un entero M, decidir si la suma de todos los elementos del vector es igual a M.*
- B=*Dado un conjunto de N enteros y un entero M, decidir si existe un subconjunto tal que la suma de los elementos del subconjunto es igual a M.*
- C=*Dado un conjunto de enteros, decidir si existe un subconjunto S tal que la suma de elementos del conjunto S es igual a la suma de los elementos que no están en S.*

a) Proponed un testigo por cada problema.
b) Dad un algoritmo en pseudocódigo que resuelva cada problema y explicad la complejidad temporal de cada algoritmo.
c) Según los testigos del primer apartado y los algorítmos del segundo, clasificad los tres problemas como P o NP.

Solución:

a) 1) El problema es polinómico y su testigo es el vector de N enteros, hay que sumar todos sus elementos y comprobar si la suma es M.

2) El testigo es un subconjunto del vector de N enteros. Hay que comprobar que la suma de los elementos de este subconjunto es igual a M.

3) El testigo también es un subconjunto del vector. Hay que comprobar que la suma de los elementos del subconjunto $\sum S = \sum S^c$.

b) Vamos a usar la función *partes(S)* que devuelve el conjunto de partes del conjunto S. El conjunto de partes es el conjunto de todos los subconjuntos formados por todos los valores de S y el conjunto vacío. Si el conjunto S tiene tamaño n, entonces su conjunto de partes tiene cardinal 2^n. Por lo tanto, la complejidad temporal de la función *partes(S)* es exponencial. Un ejemplo de lo que devuelve la función *partes(S)* sería con $S = \{1, 2, 3\}$, $\{\{1, 2, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \{\emptyset\}\}$. Por lo tanto, con $n = 3$ devuelve un conjunto con $2^3 = 8$ elementos.

1)
```
inicio
  S ← conjunto
  M ← entero
  si suma(S) = M entonces
                retorno SI
           sino
                retorno NO
  finsi
```

2)
```
inicio
  S ← conjunto
  M ← entero
  para s ← partes(S)
      si suma(s) = M
         retorno SI
      finsi
  finpara
  retorno NO
```

3) Usamos la función *complementario(s,S)* que dado un conjunto $s \in S$ devuelve el conjunto complementario s^c, tal que $s \cup s^c = S$.

```
inicio
  S ← conjunto
  para s ← partes(S)
      si suma(s) = suma(complementario(s, S))
         retorno SI
      finsi
  finpara
  retorno NO
```

c) Los tres problemas pueden verificarse en tiempo polinómico, por lo tanto son problemas *NP*. El primer problema es P ya que tenemos un algoritmo que lo resuelve con una complejidad lineal. Los otros son NP ya que sólo los podemos resolver con una complejiad exponencial.

5. (Valoración de un =10 %=5 %+5 %)
Describimos el problema *CAMIONES* de la siguiente manera. Una compañia de transporte quiere reducir su actividad comercial, abandonado algunas rutas y así optimizar costes.

Actualmente sus rutas incluyen 1000 polígonos industriales, y para cada pareja de polígonos sabe el número diario de camiones que circulan entre esos dos polígonos directamente conectados. (en los dos sentidos).

La empresa obtiene el máximo beneficio y el mínimio coste cuando, para un subconjunto de polígonos, está abasteciendo todas la rutas directas posibles entre estos polígonos con al menos 10 camiones diarios. La dirección quiere saber si actualmente hay 50 polígonos que cumplan este requisito, para limitarse a transportar mercancías entre ellos, dejando de abastecer al resto.

a) A partir de la información en el enunciado, explicad como podemos construir el grafo que representa el problema.

b) Relacionad el problema con alguno de los problemas de teoria de grafos que hemos visto en la asignatura.

Solución:

a) Cada polígono es un vértice del grafo y cada vértice o polígono está conectado con una arista cuando existe una ruta entre los dos nodos y el número de caminoes diarios entre los dos vértices es superior a 10.

b) El mejor benefício para la empresa se obtiene en los cliques del grafo con $k = 50$, por lo tanto, escojeremos el problema CLIQUE.

Recursos

Recursos Básicos

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

Recursos Complementarios

- PECs y exámenes de semestres anteriores.
- Programario para el estudio de algoritmos sobre grafos.
- Enlaces: Applets interactivos sobre algoritmos de grafos.

Criterios de valoración

- La PEC se tiene que resolver **de forma individual**.
- 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.

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**.

Este documento se tiene que entregar en el espacio **Entrega y Registro de EC** del aula **antes de las 23:59 del día 23/05/2017**. **No se aceptarán entregas fuera de término.**