PEC1

Presentación

Esta PEC es una introducción a la teoría de grafos que cubre los contenidos estudiados en los 3 primeros módulos de la asignatura. Los ejercicios trabajan tanto los conceptos previos sobre funciones y algoritmos, los fundamentos de la teoría de grafos y los problemas de recorridos y conectividades sobre grafos.

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 adecuada en cada situación y aplicar las habilidades y conocimientos adquiridos para resolverlos.

Objetivos

Los objetivos concretos de esta PEC son:

- Conocer el concepto de complejidad temporal y espacial de un algoritmo.
- Conocer el concepto de grafo y los diferentes tipos de grafos (grafos orientados, grafos ponderados, pseudografos, multigrafos, ...).
- Conococer las principales propiedades de los grafos y saber analizarlas.
- Conocer los problemas de conectividad más usuales sobre grafos, los algoritmos que los resuelven y saber aplicarlos en un grafo concreto.
- Ser capaz de representar y analizar un problema en términos de la teoría de grafos.

Descripción de la PEC a realizar

1. (Valoración de un 20 %)

   La mejor pizzeria *4 Ingredienti* de Nápoles sirve pizzas y calzones de 4 ingredientes. En el caso de la pizza, el orden en el que se ponen los ingredientes hace que el resultado de la pizza sea diferente, mientras que en el caso de la calzone, una vez se han añadido los ingredientes y doblado por la mitad, el orden en el que se añaden los ingredientes no importa. En los dos casos se contempla que los 4 ingredientes no hace falta que sean diferentes y por lo tanto hay la posibilidad de que se añada el mismo ingrediente más de una vez (incluso podemos pedir una pizza con 4 veces el mismo ingrediente). En la pizzeria tienen 10 ingredientes diferentes entre los que los clientes pueden elegir.

   a) ¿Cuántas pizzas se pueden llegar a hacer?

   b) ¿Cuántas pizzas tienen exáctamente 4 ingredientes (diferentes)?

   c) ¿Cuántas calzones se pueden llegar a servir?

   d) ¿Cuántas calzones tienen sólo dos tipos de ingredientes diferentes?

Solución:

a) Cada pizza es una secuencia de cuatro ingredientes de entre 10 posibles. Por tanto, $10^4 = 10000$.

b) Para el primer ingrediente podemos elegir de entre un total de 10, para el segundo de entre 9, y así sucesivamente. Por lo tanto, $10 \cdot 9 \cdot 8 \cdot 7 = 5040$.

c) El orden de los 4 ingredientes de la calzone son indistinguibles, así se trata de combinaciones con repetición, $CR(10, 4) = \binom{10+4-1}{4} = \binom{13}{4} = 715$.

d) $3 \cdot \binom{10}{2} = 45$, pues hay que tener en cuenta los posibles ingredientes que se eligen y cuantas veces que elige cada uno de los dos ingredientes (3 el primero y una el segundo, dos cada uno, o una vez el primero y 3 el segundo).

---

2. (Valoración de un 20 %)

   Considerad un polinomio $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ de grado n, donde los coeficientes son valores reales y a un valor real. Considerad el siguiente algoritmo:

```
función | OperacionPolinomio | ([a_n, ..., a_0], n, a)
  inicio
    poli ← a_n
    para i ← n − 1 hasta 0
        poli = poli ∗ a + a_i
        i ← i − 1
    finpara
    retorno (poli)
  fin
```

   a) Calculad $OperacionPolinomio([3, 2, 1], 2, 4)$ y $OperacionPolinomio([1, 2, 3], 2, 4)$.

   b) Razonad, dado un polinomio cualquiera de grado n $p(x) = a_n x^n + \cdots + a_1 x + a_0$, ¿qué calcula el algoritmo para un cierto valor real a?.

   c) Calculad la complejidad del algoritmo.

   d) Encontrad otro algoritmo que permita realizar el mismo cálculo. Comparad las complejidades.

Solución:

a) El resultado es $p(4) = 3 * 4^2 + 2 * 4 + 1 = 57$ y $p(4) = 1 * 4^2 + 2 * 4 + 3 = 27$, respectivamente.

b) Calcula $p(a)$. Es el conocido como *algoritmo de Horner*.

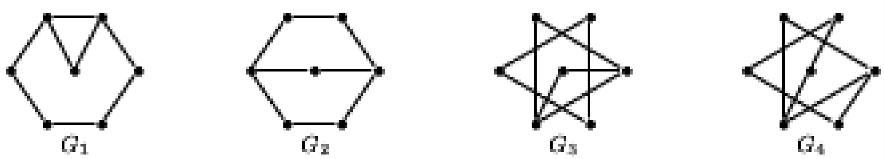
c) Se han de calcular n sumas, n restas y n productos, es decir, tiene complejidad $O(n)$.

d) Un algoritmo podria ser:

función $AvaluacionPolinomioClasica([a_n, \ldots, a_0], n, a)$
  inicio
    $poli \leftarrow a_0, eval \leftarrow 1$
    para $i \leftarrow 1$ **hasta** n
      $eval = eval \cdot a$
      $poli = poli + a_i \cdot eval$
      $i \leftarrow i + 1$
    finpara
    retorno $(poli)$
  fin

Este algoritmo en concreto realiza $2n$ multiplicaciones y $2n$ sumas, por lo que realiza el mismo número de sumas pero el doble de multiplicaciones, por lo que al igual que el algoritmo de Horner tiene complejidad $O(n)$.

3. (Valoración de un 20 %)
Considerad los siguientes grafos:



y responded razonadamente a las siguientes preguntas:

a) ¿Cuál es el número de vértices de G_i, para $i = 1, 2, 3, 4$?

b) ¿Cuál es el número de aristas de G_i, para $i = 1, 2, 3, 4$?

c) ¿Cuántos vértices de grado 2 y de grado 3 tienen cada uno de los grafos?

d) ¿Cuál de las siguientes parejas de grafos, G_3 y G_4, G_1 y G_4, y G_2, y G_3, son isomorfos?

e) ¿Qué grafos son conexos? Si alguno no lo es, ¿cuántas componentes conexas tiene?

Solución:

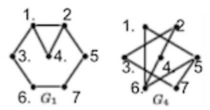
a) $| V_i | = 7$, $\forall i = 1, 2, 3, 4$.

b) $| E_i | = 8$, $\forall i = 1, 2, 3, 4$.

c) $[2, 2, 2, 2, 2, 3, 3]$ para todos los grafos.

d) 5 vértices de grado 2 y 3 de grado 3.

e) G_1 y G_4 son isomorfos. En efecto, supongamos que los vértices de los grafos están etiquetados de la sjgüiente manera:



entonces podemos construir un isomorfismo $\varphi : G_1 \to G_4$ tal que $\varphi(1) = 6$, $\varphi(2) = 5$, $\varphi(3) = 4$, $\varphi(4) = 1$, $\varphi(5) = 7$, $\varphi(6) = 2$ i $\varphi(7) = 3$ conserva las adyacencias y las no-adyacencias. En cambio, las otras parejas de grafos no lo son. Así,los grafos G_3 y G_4 no conservan las adyacencias mientras que en la pareja de grafos G_2 y G_3 un grafo es conexo mientras que el otro no lo es.

f) G_1, G_2 y G_4 son conexos y G_3 tiene dos componentes conexas.

---

4. (Valoración de un 20 %)

Si G es un grafo simple de 66 aristas, ¿cuál es el número mínimo de vértices que ha de tener G?

Solución:

Sea $\mid V \mid = n$. Como G es simple, $gr(v) \leq n-1$, $\forall v \in V$, y aplicando el lema de las encajadas tenemos que;

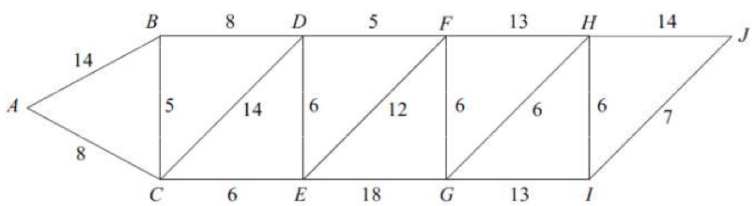
$$2 \mid A \mid = \sum_{v \in V} gr(v) \to 2 \cdot 66 = 132 \leq n(n-1).$$

Por lo tanto $0 \leq n^2 - n - 132 = (n-12)(n+11)$, de donde se puede deducir que $n \geq 12$ o bien $n \leq -11$, y como n ha de ser un número positivo, $n \geq 12$. Así el número mínimo de vértices es 12.

---

5. (Valoración de un 20 %)

El siguiente grafo muestra el tiempo, en minutos, que tarda un autobús entre los 10 pueblos de su recorrido:



a) ¿Qué algoritmo utilizaríais para encontrar el recorrido mínimo entre los pueblos A y J?

b) ¿Cuál es la ruta y qué tiempo tarda en recorrerla?

c) ¿Cuál es la ruta que tarda menos tiempo en conectar los pueblos B y J pasando por el pueblo G? Explicad razonadamente qué algoritmo utilizáis.

Solución:

a) Dijkstra.

b) Si aplicamos Dijkstra al grafo obtendremos la siguiente tabla:

A	B	C	D	E	F	G	H	I	J
(0,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)
(0,0)	(14,A)	(8,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)
(0,A)	(13,C)	(8,A)	(22,C)	(14,C)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)
((0,A)	(13,C)	(8,A)	(21,B)	(14,C)	(∞,A)	(∞,A)	(∞,A)	(∞,A)	(∞,A)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(26,E)	(32,E)	(∞,A)	(∞,A)	(∞,A)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(32,E)	(∞,A)	(∞,A)	(∞,A)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(31,F)	(38,F)	(∞,A)	(∞,A)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(31,F)	(37,G)	(44,G)	(∞,A)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(31,F)	(37,G)	(43,H)	(51,H)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(31,F)	(37,G)	(43,H)	(50,I)
((0,A)	(13,C)	(8,A)	(20,E)	(14,C)	(25,D)	(31,F)	(37,G)	(43,H)	(50,I)

que nos permite reconstruir la ruta $A - C - E - D - F - G - H - I - J$ para la que se tarda 50 minutos.

c) Aplicarem l'algorisme de Floyd, o bé l'algorisme de Dijkstra amb punt d'inici al vèrtex B. En cualquier caso obtendremos que la ruta será $B - D - F - G - H - I - J$, que se puede realizar en 38 minutos.

Recursos

Recursos Básicos

- Módulo didáctico 1. Conceptos previos: funciones y algoritmos.
- Módulo didáctico 2. Fundamentos de grafos.
- Módulo didáctico 3. Recorridos y conectividad.
- 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**.
- Cada ejercicio tiene un peso del 20 % de la nota final.
- 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: **PEC1_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 01/04/2015**. **No se aceptarán entregas fuera de término.**