Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicación

ASIGNATURA: Grafos y Complejidad

Final 1

1. (Valoración de un 5+5+5+5+5=25 %)

   El siguiente algoritmo transforma una secuencia de bits de longitud n ($n \geq 1$) en un número decimal.

```
función CambioBase10(L)
   inicio
      n ← Longitud(L)
      resultado ← 0
      para i = 1 hasta n
            si L[i] = 1 entonces
                  resultado ← resultado + 2^(n−i)
            finsi
      finpara
      retorno resultado
   fin
```

   Justificad cuáles de las siguientes afirmaciones son correctas:

   a) El resultado de la llamada CambioBase10([1,1,0,0,1]) es 19.

   b) Si $L = [1, 1, 0, 0, 1]$, la complejidad del algoritmo es $O(32)$.

   c) La complejidad del algoritmo, en función del número de bits n, es $O(n)$.

   d) El número máximo de operaciones se efectúa cuando el parámetro de entrada L tiene el máximo número de unos.

   e) Si el algoritmo sólo acepta secuencias con un número de unos fijado, igual a 5, la complejidad del algoritmo, en función de n sería $O(n)$.

   Solución:

   a) Falsa. El resultado de la llamada CambioBase10([1,1,0,0,1]) es 25.

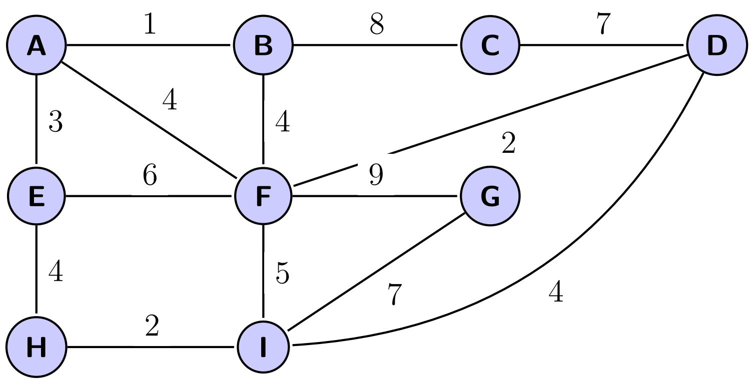
   b) Falsa. La complejidad es una medida asintótica y no tiene sentido calcularla para un valor concreto de la entrada.

c) Falsa. La complejidad es $O(n(n+5)/2+4) = O(n^2)$.

d) Cierta. Efectivamente, el número máximo de operaciones se efectúa cuando $L[i] = 1$ para todo $i = 1, \ldots, n$, y esto corresponde al caso en que la secuencia contiene todo unos.

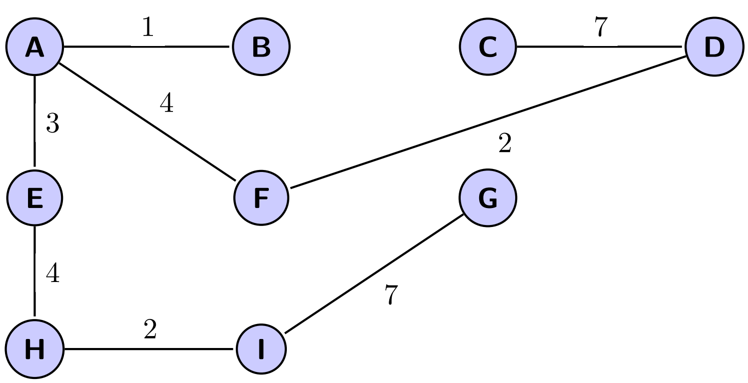
e) Cierta. Efectivamente, en este caso la complejidad sería $O(n)$, ya que el número de unos es un valor fijado igual a 5.

2. (Valoración de un 15+10=25 %) Encontrad un árbol generador minimal del siguiente grafo:



Calculad también su coste e indicad si el árbol generador minimal es único.

Solución: Aplicando el algoritmo de Kruskal, obtenemos el árbol generador minimal siguiente:



El árbol generador minimal tiene un peso de 30. Podíamos haber obtenido un árbol generador minimal diferente. Observad que el algoritmo ha elegido la arista $\{A, F\}$ que tiene peso 4 pero también podíamos haber elegido la arista $\{B, F\}$ con el mismo peso. Por lo tanto, según el orden con el que hubiéramos elegido las aristas en el algoritmo de Kruskal podríamos haber obtenido un árbol minimal diferente.

3. (Valoración de un 15+10=25 %)

   Considerad la secuencia $s: \ n, 3, 3, 2, 2, 1, 1, 1, 1$.

   a) Determinad para qué valores $1 \leq n \leq 5$ obtenemos una secuencia gráfica usando el algoritmo de Havel-Hakimi.

   b) ¿Cuál es el tamaño mínimo de un grafo con secuencia de grados s para ser conexo? ¿Para qué valor(es) de n, s es la secuencia de un grafo con esta medida?

   Solución:

   a) Por la fórmula de los grados, sabemos que el número de vértices de grado impar debe ser par, por lo tanto n debe ser par. Así, sólo debemos analizar dos casos: $n = 2$ y $n = 4$:

$n = 2$	$n = 4$
$3, 3, 2, 2, 2, 1, 1, 1, 1$	$4, 3, 3, 2, 2, 1, 1, 1, 1$
$2, 2, 1, 1, 1, 1, 1, 1$	$2, 2, 1, 1, 1, 1, 1, 1$
$1, 1, 1, 1, 1, 1, 0$	$1, 1, 1, 1, 1, 1, 0$
$1, 1, 1, 1, 0, 0$	$1, 1, 1, 1, 0, 0$
$1, 1, 0, 0, 0$	$1, 1, 0, 0, 0$
$0, 0, 0$	$0, 0, 0$
És gràfica	És gràfica

   b) Todo grafo conexo debe cumplir $|A| \geq |V| - 1 = 8$. Por lo tanto, el tamaño mínimo será $|A| = 8$. Por la fórmula de los grados tenemos que $2|A| = n+3+3+2+2+1+1+1+1 = 14+n$. Por lo tanto, $16 = 14 + n$, de donde obtenemos que para $n = 2$, la secuencia s corresponde a una grafo con el mínimo número de aristas.

4. (Valoración de un 6.25+6.25+6.25+6.25=25 %) Decid si son ciertas o falsas las afirmaciones siguientes, justificando la respuesta:

   a) El problema "Determinar si un grafo G tiene menos de 47 aristas" es un problema de cálculo.

   b) Si $A \leq_p B$, entonces no puede ser que $B \leq_p A$.

   c) Determinad si un grafo dado tiene un recorrido euleriano es un problema de complejidad polinómica.

   d) Si existe un problema A tal que $SAT \leq_p A$ y $A \in P$, entonces P=NP.

   Solución:

   a) Falso. Es un problema decisional, ya que la solución es SÍ o NO.

b) Falso, A y B podrían ser problemas polinómicamente equivalentes.

c) Cierto, porque sólo es necesario obtener la paridad de los vecinos de cada vértice, y contar cuantas veces esta paridad es impar.

d) Cierto, ya que implicaría que $SAT \in P$, y sabemos (por el teorema de Cook) que SAT es NP-completo.

Final 2

1. (Valoración de un 10+10+5=25 %)

   a) Siete personas trabajan en unas oficinas y cada uno tiene que escoger qué ordenador de sobremesa quiere tener. Disponemos de un catálogo con 6 modelos diferentes. ¿Cuántos pedidos diferentes se pueden hacer?

   b) Suponemos que en la empresa hay un total de n personas que tienen que escoger n ordenadores del catálogo anterior. Definid la función $Pedido : \mathbb{N} \rightarrow \mathbb{N}$ que, dado un valor n, nos devuelva el número de pedidos diferentes que se pueden realizar. Justificad si la función *Pedido* es o no una función biyectiva.

   c) Determinad un algoritmo que implemente la función *Pedido* i determinad su complejidad. **Nota: Podéis utilizar funciones auxiliares si lo necesitáis.**

   Solución:

   a) Se trata de una 7-muestra con repetición de un conjunto de 6 elementos, por tanto, el número de posibles pedidos es $VR(6,7) = 6^7$.

   b) La función es $Pedido(n) = 6^n$. Esta función es inyectiva ja que si $Pedido(n_1) = Pedido(n_2)$, entonces tenemos que $6^{n_1} = 6^{n_2}$ y, por tanto, $n_1 = n_2$. Ahora bien, no es exhaustiva ya que sólo las potencias de 6 tienen antiimagen; por ejemplo, 4 no tiene antiimagen ya que no existe $n \in \mathbb{N}$ tal que $6^n = 4$.

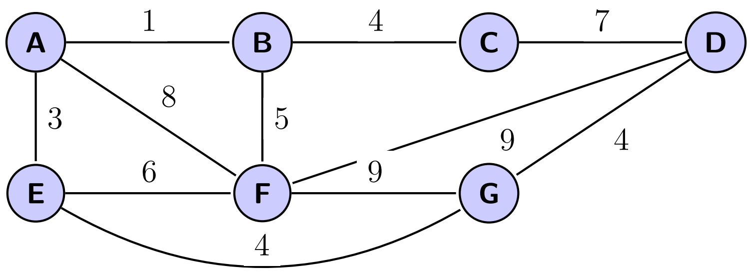
   c) Para calcular 6^n podemos utilizar el algoritmo de multiplicar y elevar que tiene un número máximo de operaciones $T(n) = \lfloor 2\log_2(n) \rfloor$.

      <u>función</u> *Pedido*(n)
        <u>inicio</u>
          <u>retorno</u> ($MultiplicarElevar(6, n)$)
        <u>fin</u>

      Por tanto, el número total de operacions se corresponde con el del algoritmo de multiplicar y elevar que tiene una complejidad de $O(\log n)$.

2. (Valoración de un 10+5+5+5=25 %)

   Considerad el siguiente grafo:



a) Encontrad la distancia mínima entre el vértice A y el vértice G del grafo anterior.

b) Recuperad el camino de coste mínimo a partir de la tabla elaborada para resolver el apartado a).

c) A partir de la tabla del apartado a), ¿podemos recuperar también el camino mínimo del vértice A al vértice D? ¿Y del vértice A al vértice F? Justificad la respuesta.

d) ¿Qué algoritmo aplicaríamos (no es necesario aplicar el algoritmo) si queremos calcular el diámetro del grafo anterior? Justificad la respuesta.

Solución:

a) Aplicando el algoritmo de Dijkstra comenzando por el vértice A se obtiene la siguiente tabla:

A	B	C	D	E	F	G
$(0, A)^*$	(∞, A)	(∞, A)	(∞, A)	(∞, A)	(∞, A)	(∞, A)
$(0, A)$	$(1, A)^*$	(∞, A)	(∞, A)	$(3, A)$	$(8, A)$	(∞, A)
$(0, A)$	$(1, A)$	$(5, B)$	(∞, A)	$(3, A)^*$	$(6, B)$	(∞, A)
$(0, A)$	$(1, A)$	$(5, B)^*$	(∞, A)	$(3, A)$	$(6, B)$	$(7, E)$
$(0, A)$	$(1, A)$	$(5, B)$	$(12, C)$	$(3, A)$	$(6, B)^*$	$(7, E)$
$(0, A)$	$(1, A)$	$(5, B)$	$(12, C)$	$(3, A)$	$(6, B)$	$(7, E)^*$

Por tanto, la distancia mínima es 7.

b) El camino desde el vértice A es $A - E - G$.

c) No podemos recuperar el camino del vértice A al vértice D, ya que el vértice D todavía no ha sido seleccionado como pivote. En cambio, sí que podemos recuperar el camino del vértice A al vértice F porque F sí ha sido seleccionado como pivote, y la distancia mínima y el camino sabemos que es 6 y $A - B - F$, respectivamente.

d) Para encontrar el diámetro del grafo, podemos aplicar primero el algoritmo de Floyd al grafo. El diámetro corresponde al máximo entre todas las distancias mínimas obtenidas con el algoritmo.

3. (Valoración de un 10+5+10=25 %)

   a) Determinad si el grafo simple $G(V, A)$ con $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ y $A = \{(1, 2), (1, 3), (2, 5), (3, 5), (3, 7), (4, 5), (4, 7), (6, 7), (8, 9), (7, 9), (6, 8)\}$ es bipartito. Justificad la respuesta. ¿Qué algoritmo aplicaríamos en general para determinar si un grafo es bipartito?

   b) Determinad, justificando la respuesta, si el grafo simple anterior es hamiltoniano.

   c) ¿Es cierto que si un grafo es euleriano, su complementario también lo será? Justificad la respuesta.

   Solución:

   a) Sí, se puede hacer una partición de V en dos subconjuntos, V_1 y V_2, de forma que toda arista relacione un vértice de V_1 y uno de V_2. Podemos tomar, por ejemplo, $V_1 = \{1, 5, 7, 8\}$ y $V_2 = \{2, 3, 4, 6, 9\}$. Podemos aplicar el algoritmo BFS o DFS etiquetando los vértices según pertenezcan a V_1 o V_2.

   b) Una condición necesaria para que un grafo bipartito sea hamiltoniano, es $|V_1| = |V_2|$. Como $|V_1| \neq |V_2|$, podemos asegurar que el grafo no es hamiltoniano.

   c) Esta afirmación no es cierta. Por ejemplo, el grafo C_6 es euleriano ya que todos sus vértices tienen grado par igual a dos, pero su complementario tiene todos los vértices de grado 3, y por tanto no puede ser euleriano.

4. (Valoración de un 6.25+6.25+6.25+6.25=25 %) Decid si son ciertas o falsas las afirmaciones siguientes, justificando la respuesta:

   a) Todo problema de cálculo es un problema de optimización.

   b) Si la entrada de un algoritmo es el número entero 78 (en base 10), el tamaño de la entrada es 7.

   c) El problema "Determinar si un grafo es 3-colorable" pertenece a NP.

   d) Si los problemas A y B son NP-completos, entonces $A \leq_p B$ y $B \leq_p A$.

   Solución:

a) Falso. Es al revés, todo problema de optimización es un problema de cálculo.

b) Cierto, porque su representación binaria, 1001110, consta de 7 bits.

c) Cierto, ya que si nos dan una posible coloración como testimonio podemos comprobar que es correcta en tiempo polinómico.

d) Cierto. Dada una clase de complejidad C, todos los problemas que son C-completos son polinómicamente equivalentes.

Final 3

1. (Valoración de un 15+10=25 %)

   a) Considerad el siguiente algoritmo que retorna el término n de la sucesión de Fibonacci, para $n > 2$.

```
1  función Fibonacci(n)
2    inicio
3      F1 ← 0
4      F2 ← 1
5      F3 ← 1
6      para i = 3 hasta n
7          F3 ← F1 + F2
8          F1 ← F2
9          F2 ← F3
10     finpara
11     retorno (F3)
12   fin
```

   ¿Cuántas operaciones realiza el algoritmo cuando hace la llamada a $Fibonacci(n)$? Determinad la complejidad del algoritmo en función de n.

   b) Sabemos que el término n-ésimo de otra sucesión viene dado por la expresión $S_n = \frac{1}{2} \cdot 3^n - 2^n$. Diseñad un algoritmo que retorne S_n utilizando la fórmula anterior e indicad qué complejidad tiene. **Nota: Podéis utilizar funciones auxiliares si lo necesitáis.**

Solución:

a) Las líneas 3, 4 y 5 efectúan una operación elemental cada una. La línea 6 efectúa una inicialización, $n-1$ comparaciones y $n-2$ incrementos de la variable, por tanto un total de $2n-2$ operaciones elementales. La línea 7 efectúa una operación elemental con la suma, una con la asignación y las línias 8 y 9 una operación elemental. El número de iteraciones es $n-2$, y por tanto el total de operaciones elementales durante el bucle es $4(n-2)$.

Por tanto, el número total de operaciones elementales será, $3 + 2n - 2 + 4(n-2) = 6n - 7$. De acuerdo con las propiedades de la complejidad, este algoritmo tendrá una complejidad $O(n)$.

b) Para calcular las potencias podemos utilizar el algoritmo de multiplicar y elevar que tiene un número máximo de operaciones $T(n) = \lfloor 2 \log_2(n) \rfloor$.

```
1 función Sucesion(n)
2   inicio
3     a ← MultiplicarElevar(3, n)
4     b ← MultiplicarElevar(2, n)
5     retorno (½a − b)
6   fin
```

Las líneas 3 y 4 realizan $\lfloor 2\log_2(n) \rfloor + 1$ operaciones cada una; las del algoritmo de multiplicar y elevar y la de la asignación. Finalmente, la línea 5 realiza una división, un producto y una resta. Por tanto, el número total de operaciones es $T(n) = 2\lfloor 2\log_2(n) \rfloor + 5$ y el algoritmo tiene una complejidad $O(\log n)$.

2. (Valoración de un 15+10=25 %)

   a) Dibujad el árbol correspondiente a la expresión aritmética

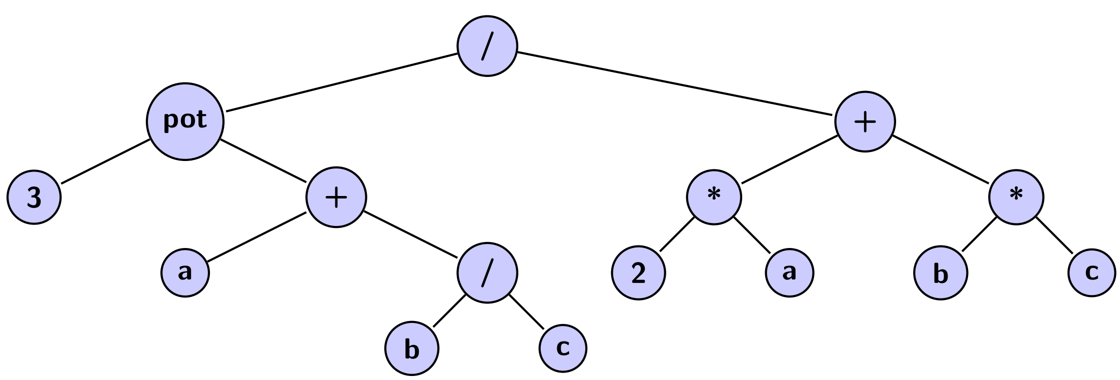
   $$\frac{3^{a+\frac{b}{c}}}{2a+bc}$$

   teniendo en cuenta la prioridad habitual de los operadores.

   b) Determinad el recorrido en preorden, inorden y postorden del árbol del apartado anterior.

Solución:

a) El árbel es



b) Estos son los recorridos:

- Preorden: /,pot,3,+,a,/,b,c,+,*,2,a,*,b,c;

- Inorden: 3,pot,a,+,b,/,c,/,2,*,a,+,b,*,c;
- Postorden: 3,a,b,c,/,+,pot,2,a,*,b,c,*,+,/.

3. (Valoración de un 6.25+6.25+6.25+6.25=25 %) Aplicando el algoritmo de Dijkstra a un grafo ponderado de 6 vértices obtenemos la siguiente tabla:

A	B	C	D	E	F
$(0, A)$	(∞, A)	(∞, A)	(∞, A)	(∞, A)	$(\infty, 0)$
$(0, A)$	$(2, A)$	$(1, A)$	(∞, A)	(∞, A)	$(4, A)$
$(0, A)$	$(2, A)$	$(1, A)^*$	$(2, C)$	(∞, A)	$(4, A)$
$(0, A)$	$(2, A)^*$	$(1, A)$	$(2, C)$	$(4, B)$	$(4, A)$
$(0, A)$	$(2, A)$	$(1, A)$	$(2, C)^*$	$(3, D)$	$(4, A)$
$(0, A)$	$(2, A)$	$(1, A)$	$(2, C)$	$(3, D)^*$	$(4, A)$
$(0, A)$	$(2, A)$	$(1, A)$	$(2, C)$	$(3, D)$	$(4, A)$

Justificad si son ciertas o falsas las siguientes afirmaciones:

a) El grafo es euleriano.

b) El diámetro del grafo es 4.

c) El coste de un árbol generador minimal es inferior o igual a 9.

d) El coste de la arista $\{E, B\}$ es 4.

Solución:

a) Falso. El vértice A tiene grado 3 y, por tanto, el grafo no puede ser euleriano.

b) Falso. En la tabla, 4 es el máximo de las distancias mínimas entre los vértices A y el resto de vértices. Esto no nos permite asegurar que la distancia entre otros vértices no pueda ser más grande.

c) Cierto. Si consideramos las aristas $\{\{A, B\}, \{A, C\}, \{C, D\}, \{D, E\}, \{A, F\}\}$ tenemos un árbol generador de coste 9. Por tanto, un árbol generador minimal tiene que tener coste menor o igual a 9.

d) Falso. Según podemos ver en la segunda fila, el coste de la arista $\{E, B\}$ es 4 menos el coste de la arista $\{A, E\}$, que es 2; por tanto, el coste de la arista $\{E, B\}$ es 2.

4. (Valoración de un 6.25+6.25+6.25+6.25=25 %) Decid si son ciertas o falsas las afirmaciones siguientes, justificando la respuesta:

a) La fórmula booleana siguiente está en FNC (forma normal conjuntiva): $(a \vee b) \wedge (\overline{b} \vee c) \wedge \overline{a}$

b) Si un algoritmo para resolver un problema *Prob* tiene complejidad (temporal) exponencial, entonces *Prob* es intratable.

c) Si un problema A es NP-completo, entonces A es NP-difícil.

d) Si A es un problema NP-completo y $B \in NP$, entonces $B \leq_p A$.

Solución:

a) Cierto. Es una conjunción de cláusulas.

b) Falso, la complejidad temporal será, como máximo, exponencial, pero podría ser inferior.

c) Cierto por definición. Además podemos afirmar que $A \in NP$.

d) Cierto, porque en particular A es NP-difícil.