

CI-0116 Análisis de Algoritmos y Estructuras de Datos
I ciclo de 2020

III EXAMEN PARCIAL

Lunes 6 de julio

El examen consta de 7 preguntas que suman 104 puntos (4% extra). Las preguntas se pueden responder en cualquier orden, pero se debe indicar en el cuadro de abajo las páginas del cuaderno de examen en la que están las respuestas. Para ello numere las hojas del cuaderno de examen en la esquina superior externa de cada página. El examen se puede realizar con lápiz o lapicero. No se permite el uso de dispositivos electrónicos (calculadoras, teléfonos, audífonos, etc.).

Pregunta	Puntos	Páginas	Calificación
1. <i>Representación de grafos</i>	10		
2. <i>Recorrido de un grafo en anchura</i>	8		
3. <i>Componentes fuertemente conexos</i>	20		
4. <i>Algoritmo de Kruskal y estructuras de datos para conjuntos disjuntos</i>	30		
5. <i>Algoritmo de Prim</i>	10		
6. <i>Camino más corto desde una fuente</i>	8		
7. <i>Camino más corto entre todos los vértices</i>	18		
Total	104		

1. *Representación de grafos* [10 pts.]

Suponga que se desea encontrar en un grafo con V vértices y A aristas el vértice que es adyacente de una mayor cantidad de vértices.

- ¿Cuánto tiempo toma encontrar tal vértice en un grafo representado mediante listas de adyacencia? Explique. [4 pts.]
- ¿Cuánto tiempo toma encontrarlo en un grafo representado mediante una matriz de adyacencia? Explique. [4 pts.]
- De acuerdo a su respuestas anteriores, ¿que representación de grafo permite encontrar el vértice que es adyacente de una mayor cantidad de vértices más rápidamente? [2 pts.]

2. *Recorrido de un grafo en anchura* [8 pts.]

Realice un recorrido en anchura de la versión *no dirigida* y sin pesos del grafo de la figura 1, partiendo del vértice a . Asuma que las listas

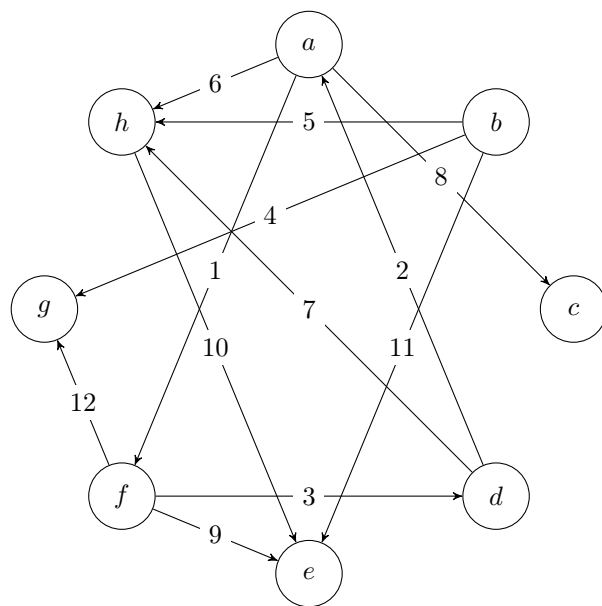


Figura 1: Grafo para las preguntas del examen. En algunas se usa una versión no dirigida, en otras una versión sin pesos, y en otras un subgrafo. Preste atención a los detalles de cada pregunta.

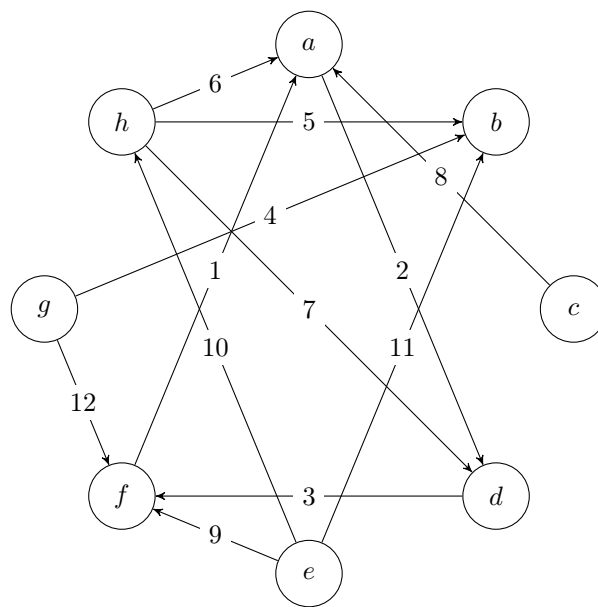


Figura 2: Traspuesta del grafo de la figura 1.

de adyacencia están ordenadas en el sentido del movimiento de las manecillas del reloj, empezando en el vértice correspondiente. Muestre al finalizar cada iteración principal del algoritmo la distancia de la fuente (δ) a cada uno de los otros vértices, los predecesores que forman el camino correspondiente (π) y el estado de la cola. Si al finalizar una iteración no muestra la distancia o el predecesor de un vértice, se asume que son los mismos de la iteración anterior.

It.	<i>a</i>		<i>b</i>		<i>c</i>		<i>d</i>		<i>e</i>		<i>f</i>		<i>g</i>		<i>h</i>		Cola
	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π	
0	0	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	<i>a</i>
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	

3. Componentes fuertemente conexos [20 pts.]

- Haga un recorrido en profundidad del grafo de la figura 1. Muestre los tiempos de descubrimiento y finalización y los predecesores de cada vértice. Asuma que en el ciclo principal los vértices se toman en el sentido del movimiento de las manecillas del reloj, empezando en *a*, y que las listas de adyacencia están ordenadas también en el sentido de las manecillas del reloj, empezando en el vértice correspondiente. [8 pts.]
- Clasifique las aristas que no son de árbol como aristas hacia adelante, hacia atrás o cruzadas, márcandolas con la letra correspondiente —*F*, *X* o *B*, respectivamente—, del lado del culatín de la flecha. [4 pts.]
- Haga un recorrido en profundidad de la traspuesta del grafo (figura 2) tomando los vértices en orden decreciente de finalización de la parte a) y asumiendo que las listas de adyacencia están ordenadas en el sentido de las manecillas del reloj, a partir del vértice correspondiente. Muestre los tiempos de descubrimiento y finalización del recorrido [4 pts.]. Use esta información para identificar los componentes fuertemente conexos del grafo [4 pts.].

4. Algoritmo de Kruskal y estructuras de datos para conjuntos disjuntos [30 pts.]

Use el algoritmo de Kruskal para encontrar un árbol recubridor mínimo de la versión no dirigida del grafo de la figura 1. Muestre para cada iteración lo siguiente:

- La arista considerada, y si fue tomada o no. (Después de la primera arista mal procesada el resto no suman puntos). [10 pts.]
- El estado de la correspondiente estructura de datos para conjuntos disjuntos, implementada mediante árboles. Indique el rango de los nodos mediante subíndices. Si gusta, puede mostrar solo los árboles que cambian. (Después del primer árbol mal procesado, el resto no suman puntos). [20 pts.]

Asuma que los argumentos de la operación UNION son dados en orden alfabético (por ejemplo, como UNION(a, b) y no UNION(b, a)). Si al finalizar una iteración no muestra el padre ni el rango de un nodo, se asume que son los mismos que en la iteración anterior (el nodo y cero, respectivamente, si no se indican en la primera iteración).

5. *Algoritmo de Prim* [10 pts.]

Use el algoritmo de Prim para encontrar un árbol recubridor mínimo de la versión no dirigida del grafo de la figura 1. Especifique en cada iteración principal del algoritmo las aristas consideradas y la arista seleccionada. Empiece en el vértice a . (Después de la primera arista mal procesada el resto de aristas no suman puntos).

6. *Camino más corto desde una fuente* [8 pts.]

Use el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice a y los otros vértices en la versión *no dirigida* del grafo de la figura 1. Al considerar un vértice como pivote, actualice las distancia y predecesores de sus adyacentes *solo si la distancia disminuye* (no si se mantiene igual). Si varios vértices tienen la misma distancia conocida a la fuente, dé prioridad al menor alfabéticamente. Muestre en cada iteración las distancias más cortas conocidas desde la fuente (δ) [$3\frac{1}{2}$ pts.] y los predecesores que forman tal camino (π) [$3\frac{1}{2}$ pts.]. Si al finalizar una iteración no indica la distancia o el predecesor de un vértice, se asume que son los mismos que en la iteración anterior.

It.	a		b		c		d		e		f		g		h	
	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π	δ	π
0	0	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL	∞	NIL
1																
2																
3																
4																
5																
6																
7																
8																

7. *Camino más corto entre todos los vértices* [18 pts.]

Simule la ejecución del algoritmo de Floyd y Warshall sobre la versión no dirigida del subgrafo formado por los vértices a, b, c, d y e de la figura 1. Si deja una casilla vacía en una matriz $D^{(k)}$ se asume que tiene un valor de infinito y si deja una casilla vacía en una matriz $\Pi^{(k)}$ se asume que tiene un valor de NIL. [$1\frac{1}{2}$ pto. cada matriz. Después de la primer matriz D incorrecta el resto de matrices D no suman puntos. Después de la primer matriz Π incorrecta el resto de matrices Π no suman puntos.]

$$D^{(0)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(0)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$D^{(1)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(1)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$D^{(2)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(2)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$D^{(3)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(3)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$D^{(4)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(4)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$D^{(5)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0

$$\Pi^{(5)} =$$

	a	b	c	d	e
a	0				
b		0			
c			0		
d				0	
e					0