

**CI-1221 Estructuras de datos y análisis de algoritmos**  
**II ciclo de 2017**

**IV EXAMEN PARCIAL**

*Jueves 30 de noviembre, 12:00 m. d. – 3:00 p. m.*

Nombre: \_\_\_\_\_ Carné: \_\_\_\_\_

El examen consta de 6 preguntas que suman  $121\frac{1}{2}$  puntos, pero no se reconocerán más de 130 (30 % extra). Se recomienda echar un vistazo a los temas de las preguntas y a su puntaje antes de resolver el examen, para así distribuir su tiempo y esfuerzo de la mejor manera. Las preguntas se pueden responder en cualquier orden, pero se debe indicar en el cuadro mostrado en la portada 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>	8		
2. <i>Recorrido de un grafo en anchura</i>	14		
3. <i>Ordto. topológico y componentes fuertemente conexos</i>	$34\frac{1}{2}$		
4. <i>Algoritmo de Kruskal y ests. de datos para conj. disjtos.</i>	22		
5. <i>Algoritmo de Prim</i>	13		
6. <i>Camino más corto desde una fuente y colas de prioridad</i>	30		
Total	$121\frac{1}{2}$		

1. *Representación de grafos* [8 pts.]

¿Qué representación del grafo de la figura 1 toma más espacio: una matriz de adyacencia o listas de adyacencia? Asuma que los vértices se identifican mediante enteros consecutivos (1, 2, etc.), que los pesos de las aristas son también enteros (negativos si no hay arista, en el caso de la matriz), que los apuntadores en las listas de adyacencia toman el mismo espacio que un entero y que las listas no son circulares ni tienen nodo centinela. Muestre sus cálculos.

2. *Recorrido de un grafo en anchura* [14 pts.]

Realice un recorrido en anchura de la versión *no dirigida* y sin pesos del grafo de la figura 1. Parta del vértice «Probabilidad y Estadística» e ignore los pesos de las aristas. Asuma que las listas de adyacencia están ordenadas alfabéticamente. Muestre al finalizar cada iteración principal del algoritmo la distancia de la fuente a cada uno de los vértices, los predecesores que forman el camino correspondiente y el estado de la cola. Si al finalizar una iteración no muestra la distancia ni el predecesor de un nodo, se asumirá que son los mismos que en la iteración anterior ( $\infty$  y NIL, respectivamente, si no se indican en la primera iteración).

3. *Ordto. topológico y componentes fuertemente conexos* [ $34\frac{1}{2}$  pts.]

- Haga un recorrido en profundidad del grafo dirigido de la figura 1 y muestre los tiempos de descubrimiento y finalización y vértices predecesores correspondientes [7 pts.]. Asuma que las listas de adyacencia están ordenadas alfabéticamente y que en el ciclo principal los vértices son tomados en orden alfabético. Ignore los pesos de las aristas.
- Indique el ordenamiento topológico del grafo inducido por el recorrido anterior [ $3\frac{1}{2}$  pts.]. (Puede usar abreviaturas siempre y cuando no haya ambigüedad).

Ciclo

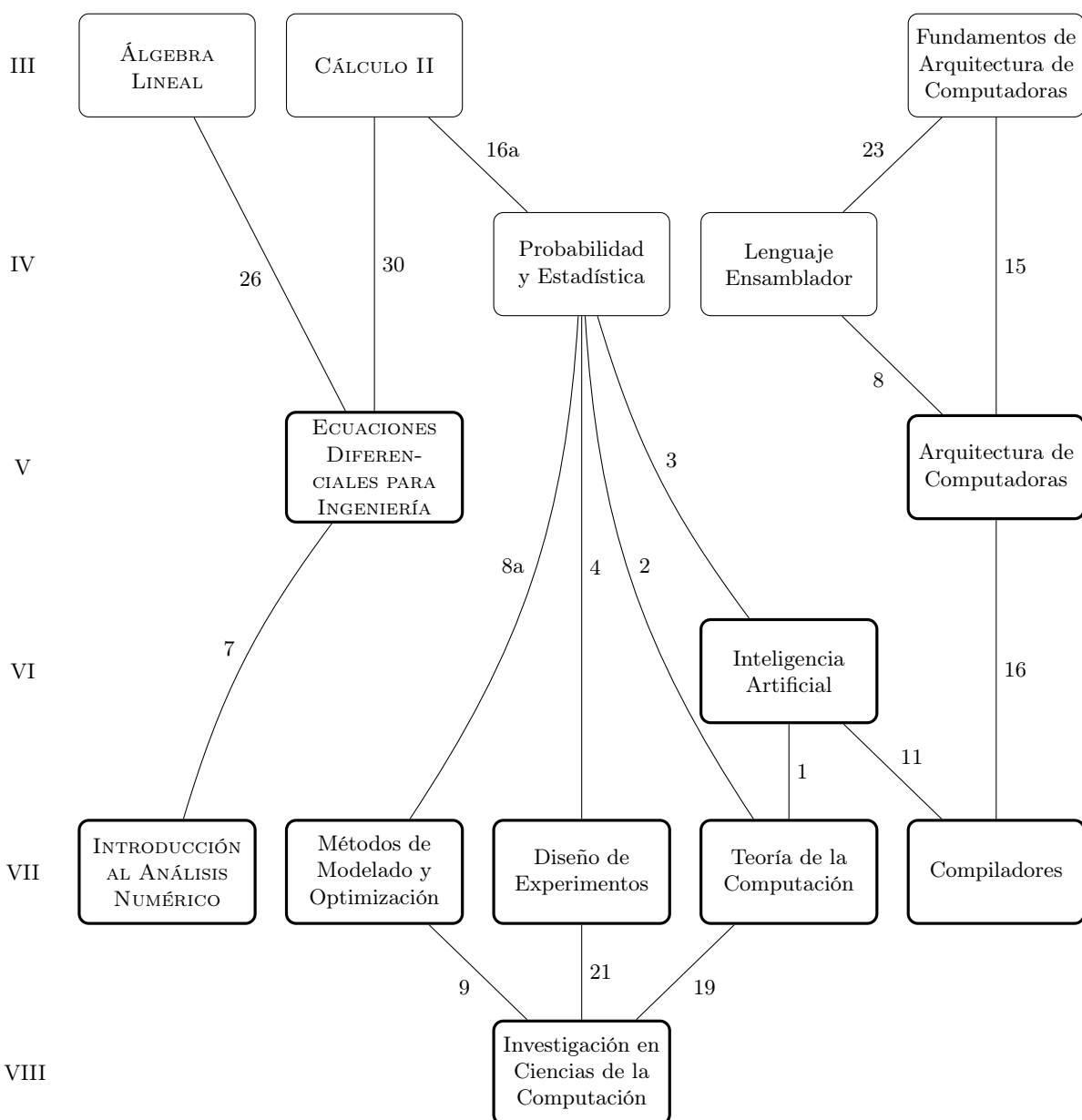


Figura 1: Requisitos de los cursos propios (con borde grueso) del nuevo énfasis de Ciencias de la Computación del bachillerato en Computación de la Universidad de Costa Rica. Los pesos de las aristas representan la diferencia entre las longitudes de los nombres de los cursos. Cuando hay pesos repetidos, a uno de ellos se le agrega la letra «a» para diferenciarlos (por ejemplo, «8» y «8a»), pero la letra no cambia su valor numérico. Este grafo y variaciones de él se usan en las preguntas 1, 2, 3, 4, 5 y 6.

- c) Etiquete las aristas que no pertenecen a árboles como aristas hacia adelante (F), hacia atrás (B) o cruzadas (X). [7 pts.]
- d) Elimine las aristas del grafo que no pertenecen a árboles y dibuje la versión no dirigida del grafo resultante [4 pts.]. Asuma que se hace un recorrido en profundidad de este grafo produciendo los tiempos de descubrimiento y finalización mostrados en la parte a). (Esto significa que las listas de adyacencia no necesariamente estaban ordenadas alfabéticamente cuando se hizo este recorrido en profundidad). Ahora haga un recorrido en profundidad de la traspuesta del grafo (que por ser no dirigido es él mismo) tomando los vértices en cada visita de acuerdo a los tiempos de finalización obtenidos en la parte a) en orden decreciente. Muestre los respectivos tiempos de descubrimiento y finalización y los predecesores [7 pts.]. Use esta información para encontrar los componentes fuertemente conexos del grafo [6 pts.].

4. *Algoritmo de Kruskal y ests. de datos para conjts. disjtos.* [22 pts.]

Use el algoritmo de Kruskal para encontrar un árbol recubridor mínimo para la versión *no dirigida* del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). Las arista etiquetada «8a» tiene un peso de 8, pero debe ser considerada después de considerar la arista etiquetada como «8» (también con peso 8); lo mismo aplica para las aristas «16a» y «16». Así, el orden en que se deben considerar las aristas es el siguiente: 1, 2, 3, 4, 7, 8, 8a, 9, 11, 15, 16, 16a, 19, 21, 23, 26 y 30. 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). [11 pts.]
- El estado de la correspondiente estructura de datos para conjuntos disjuntos, implementada mediante árboles. Indique el rango de los nodos por medio de subíndices. (Después del primer árbol mal procesado, el resto no suman puntos). [11 pts.]

Para esta pregunta asuma que los argumentos de la operación UNION son dados en orden alfabético (por ejemplo, como  $\text{UNION}(a, b)$  y no  $\text{UNION}(b, a)$ ). Si al finalizar una iteración no muestra el padre ni el rango de un nodo, se asumirá 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* [13 pts.]

Use el algoritmo de Prim para encontrar un árbol recubridor mínimo de la versión *no dirigida* del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). Especifique en cada iteración principal del algoritmo las aristas consideradas [6½ pts.] y la arista seleccionada [6½ pts.]. Empiece en el vértice ÁLGEBRA LINEAL. (Después de la primera arista mal procesada el resto de aristas no suman puntos).

6. *Camino más corto desde una fuente y colas de prioridad* [30 pts.]

Use el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice «Teoría de la Computación» y los otros vértices en la versión *no dirigida* del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). Muestre en cada iteración las distancias más cortas conocidas desde la fuente [7½ pts.], los predecesores que forman tal camino [7½ pts.] y el estado de la cola de prioridad correspondiente (con mostrar las llaves es suficiente) [15 pts.]. Si al finalizar una iteración no indica la distancia o el predecesor de un vértice, se asumirá que son los mismos que en la iteración anterior ( $\infty$  y NIL, respectivamente, si no se han especificado aún).