

CI-1221 Estructuras de datos y análisis de algoritmos
I ciclo de 2018

IV EXAMEN PARCIAL

Martes 10 de julio, 5:00 p. m. – 7:30 p. m.

Nombre: _____ Carné: _____

El examen consta de 6 preguntas que suman 110 puntos (10 % extra). 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>	10		
2. <i>Recorrido de un grafo en anchura</i>	12		
3. <i>Ordto. topológico y componentes fuertemente conexos</i>	$36\frac{1}{2}$		
4. <i>Algoritmo de Kruskal y ests. de datos para conjs. disjtos.</i>	$26\frac{1}{2}$		
5. <i>Algoritmo de Prim</i>	13		
6. <i>Camino más corto desde una fuente</i>	12		
Total	110		

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

Calcule cuánto espacio toma el grafo de la figura 1 si se representa como una matriz de adyacencia y si se representa como 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. Dé los resultados en términos del tamaño de un entero.

2. *Recorrido de un grafo en anchura* [12 pts.]

Realice un recorrido en anchura de la versión *no dirigida* y sin pesos del grafo de la figura 1. Parta del curso que es requisito para una mayor cantidad de cursos —Proyecto Integrador de Ingeniería de Software y Bases de Datos— 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 cursos, 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 asume que son los mismos que en la iteración anterior (∞ y NIL, respectivamente, si no se indican en la primera iteración).

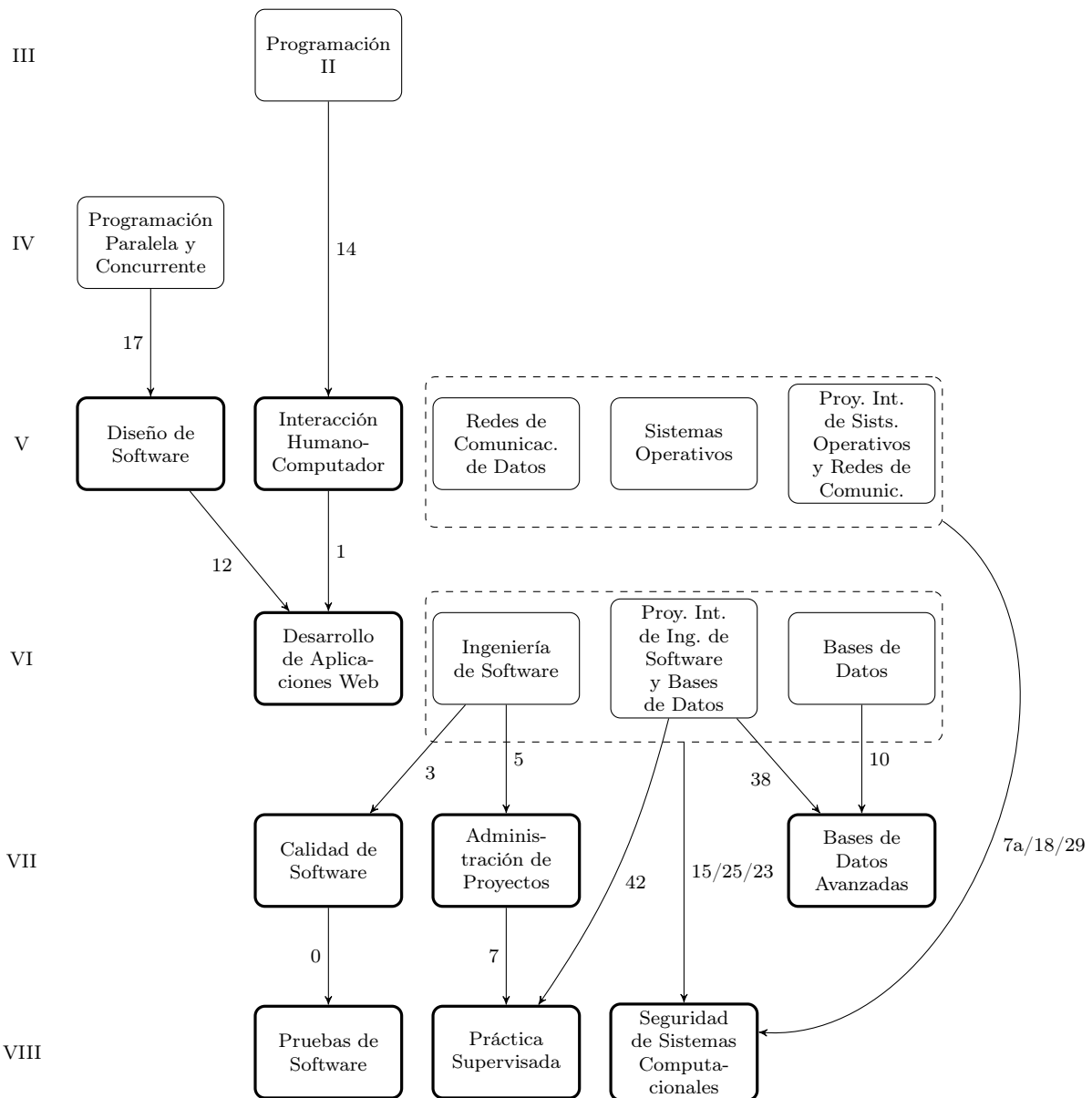


Figura 1: Requisitos de los cursos propios (con borde grueso) del nuevo énfasis de Ingeniería de *Software* del bachillerato en Computación de la Universidad de Costa Rica. Las flechas que salen de un bloque de cursos (representado mediante un agrupamiento con línea discontinua) indican que todos los cursos del bloque son requisitos del curso al que apunta la flecha. Los pesos de las aristas representan la diferencia entre las longitudes de los nombres de los cursos (sin abreviaturas e incluyendo espacios). En el caso de flechas que salen de bloques, se indican varios pesos, cada uno correspondiente a una representación equivalente en la que saldría una flecha de cada uno de los cursos del bloque. Cuando hay pesos repetidos, a uno de ellos se le agrega la letra «a» para diferenciarlos (por ejemplo, «7» y «7a»), pero esta no cambia el valor numérico.

3. *Ordo. topológico y componentes fuertemente conexos* [36½ 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. Asuma que las listas de adyacencia están ordenadas alfabéticamente y que en el ciclo principal los vértices se toman en orden alfabético. Ignore los pesos de las aristas. [8½ pts.]
- Según el ordenamiento topológico inducido por el recorrido anterior, ¿en qué orden se deben tomar los cursos? (Puede usar abreviaturas siempre y cuando no causen ambigüedad). [4 pts.]
- Etiquete las aristas que no pertenecen a árboles como aristas hacia adelante (F), hacia atrás (B) o cruzadas (X). [3 pts.]
- Elimine las aristas del grafo que no pertenecen a árboles y dibuje la versión no dirigida del grafo resultante [4 pts.]. Suponga que se hace un recorrido en profundidad de este grafo y que se producen los tiempos de descubrimiento y finalización mostrados en la parte a). Haga un recorrido en profundidad de la traspuesta del grafo (que por no ser dirigido es él mismo) tomando los vértices en cada visita según los tiempos de finalización obtenidos en la parte a) en orden decreciente. Muestre los respectivos tiempos de descubrimiento y finalización y los predecesores [8½ pts.]. Use esta información para encontrar los componentes fuertemente conexos del grafo [8½ pts.].

4. *Algoritmo de Kruskal y ests. de datos para conjs. disjtos.* [26½ 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). (La arista etiquetada «7a» tiene un peso de 7, pero debe ser considerada después de considerar la arista etiquetada como «7», también con peso 7). Así, el orden en que se deben considerar las aristas es el siguiente: 0, 1, 3, 5, 7, 7a, 10, 12, 14, 15, 17, 18, 23, 25, 29, 38 y 42. 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). [8½ 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. (Si gusta, puede mostrar solo los árboles que cambian; después del primer árbol mal procesado, el resto no suman puntos). [17 pts.]

Para esta pregunta 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 asumirá que son los mismos que en la iteración anterior (el nodo y cero, respectivamente, si no se indican en la primera iteración). *Indique si el algoritmo fue capaz de hallar un árbol recubridor* [1 pto.].

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 y la arista seleccionada [13 pts.]. Empiece en el vértice alfabéticamente menor: «Administración de Proyectos». (Después de la primera arista mal procesada el resto de aristas no suman puntos). *Indique si el algoritmo fue capaz de hallar un árbol recubridor* [1 pto.].

6. *Camino más corto desde una fuente* [12 pts.]

Use el algoritmo de Dijkstra para encontrar el camino más corto entre el vértice «Proyecto Integrador de Ingeniería de Software y Bases de Datos» y los otros vértices en la versión *no dirigida* del grafo de la figura 1 (es decir, ignorando la dirección de las flechas). *Si varios vértices tienen la misma distancia conocida a la fuente, dele prioridad al menor, en un sentido alfabético.* Muestre en cada iteración las distancias más cortas conocidas desde la fuente [6 pts.] y los predecesores que forman tal camino [6 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 (∞ y NIL, respectivamente, si no se han especificado).