

Estructuras de Datos y de la Información II

Examen final – 4 septiembre 2007

Observación: Las preguntas se deberán entregar en tres bloques:

- Bloque 1: preguntas 1 y 2.
- Bloque 2: preguntas 3 y 4.
- Bloque 3: preguntas 5 y 6.

Bloque 1

1.5 pt.

1) Dada la siguiente cadena:

ABABBBCCDEFFGGABABCBCBEBEBFBFBGBGBBDD

- a. Construir códigos de Huffman para cada símbolo de esta cadena mediante un algoritmo apropiado. Aplicar los códigos resultantes a la cadena anterior, y calcular el tamaño del texto comprimido en comparación con el original, suponiendo que cada carácter de la cadena sin comprimir ocupaba 8 bits.
- b. Aplicar la codificación “run-length” a la cadena inicial, mostrando como quedaría el texto comprimido y comparando nuevamente el tamaño antes y después de comprimir.

¿Cuál es la diferencia de compresión mediante los dos métodos? Razona todos los resultados obtenidos.

NOTA IMPORTANTE: Todos los pasos en la ejecución del algoritmo Huffman deben estar claros y explicados.

1.5 pt.

2) Demostrar mediante contraejemplos que las siguientes afirmaciones son falsas:

- a. Dado un grafo y fijando un vértice para la ejecución del algoritmo BFS, siempre se obtiene el mismo árbol BF.
- b. En DFS, si un nodo v es accesible desde otro nodo u , y v no ha sido aún descubierto en el instante en que se descubre u , entonces a la fuerza v descende de u en el bosque DF.

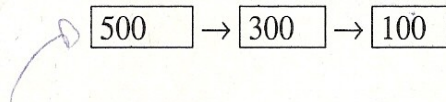
Mostrar un ejemplo de cómo:

- c. El algoritmo DFS (o alguna pequeña variante) detecta ciclos en un grafo.

NOTA IMPORTANTE: Todos los ejemplos y contraejemplos deben razonarse junto con los pasos en la ejecución de los algoritmos que se utilicen para explicarlos.

1.5 pt.

- 3) Un fichero con registros de longitud variable, con reutilización del espacio mediante lista de registros borrados contiene, en un momento dado, varios registros llenos, y tres registros borrados, organizados éstos en una lista como sigue (donde el número dentro de cada elemento representa el tamaño del registro borrado):



En este fichero se efectúan, por orden, las operaciones siguientes:

- se borra un registro de 200 bytes (y se añade a la lista de borrados).
- se inserta un registro de 90 bytes (sacando espacio de la lista de borrados).
- se inserta un registro de 100 bytes (sacando espacio de la lista de borrados).

Escribir el estado de la lista de borrados después de cada una de las operaciones precedentes en el caso en que no se creen nunca fragmentos internos y que la estrategia de gestión de la lista sea:

- best fit
- worst fit
- first fit

1.5 pt.

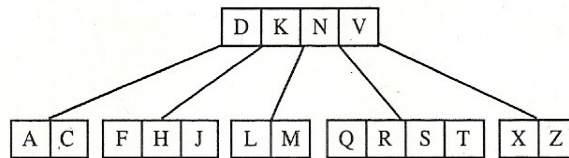
- 4) Considerando la base de datos siguiente:

<u>Nombre</u>	<u>DNI</u>	<u>Dirección</u>	<u>Ocupación</u>
Juan Jiménez	0809765	Barquillo, 3	Médico
Pablo González	1890657	Murillo, 39	Abogado
Ana López	3490768	Andalucía, 18	Abogado
Carlos Morales	3497654	Libertad, 28	Consultor

- Construir un índice primario (utilizando la clave oportuna), un índice secundario por nombre y otro por ocupación.
- En esta base de datos con índices, insertar un registro para Juana Carrillo, con DNI 2864510, dirección Serrano 34, y de profesión consultor. Especificar cómo quedan la base de datos y los índices después de la inserción.
- De esta base de datos, eliminar Juan Jiménez. Especificar cómo quedan la base de datos y los índices después de la eliminación.

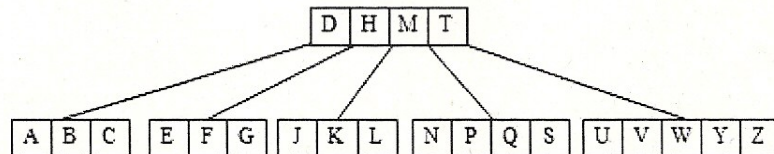
Bloque 3

2 pts. 5) Dado el siguiente árbol B, con páginas de tamaño para 4 claves:



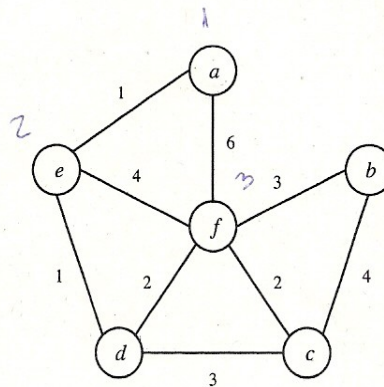
- Insertar la clave P.
- Partiendo del resultado anterior, eliminar L.
- Partiendo del resultado anterior, eliminar J.

Dado el siguiente árbol B*, con páginas de tamaño para 5 claves:



- Insertar X.
- Partiendo del resultado anterior, insertar R.

2 pt. 6) Dado el siguiente grafo:



- Encontrar los caminos de coste mínimo desde *a* hasta todos los demás, utilizando el algoritmo más eficiente posible.
- Encontrar un subgrafo que una todos los nodos, de forma que el coste total del subgrafo sea el mínimo posible.

En ambos apartados, si se utiliza alguna estructura de datos auxiliar, mostrar su estado en cada paso del algoritmo. Donde los algoritmos dejen libertad en el orden de recorrido de los nodos o arcos, éste se realizará en orden alfabético.