



Estructuras de Datos

Grado en Ingeniería en Informática
Examen de Septiembre de 2012

Nombre:

Grupo:

Pregunta 1:

Contestar V/F las siguientes cuestiones, teniendo en cuenta que una respuesta incorrecta invalida una correcta.

[V] Una lista doblemente enlazada y circular puede considerarse una eedd lineal, de acceso secuencial y dinámica.

[V] Un dato almacenado en una lista dinámica nunca cambia en su ubicación en memoria, aunque se inserten o eliminen más datos en dicha lista.

[F] Si la clase A tiene como atributo un vector de punteros a objetos tipo B, entonces entre ellos sólo puede existir una relación de asociación.

[F] Si P y Q son dos conjuntos de bits de tamaños 7 y 15 respectivamente, entonces $P \cup Q$ puede tener tamaño 10.

[V] La operación $\text{pop}()$ de un heap montado sobre un vector se puede mejorar eligiendo una lista de listas para dicha implementación del heap.

[M] Borrar un nodo de un árbol ABB de tamaño 1000 pue-

de tener tan sólo un coste de $T(1000)=2$, es decir, visitando tan sólo 2 nodos.

[V] Elisa lleva razón cuando dice que no va a usar una tabla hash para su aplicación porque, aunque necesita realizar búsquedas eficientes, también necesita realizar listados ordenados de datos.

[V] Un grafo que representa carreteras nacionales puede considerarse un grafo ponderado, no dirigido y posiblemente cíclico.

[V] Las posiciones de las gasolineras sobre un mapa se van a representar utilizando un Kd-tree. La búsqueda de dichas gasolineras en un rango $[x1,y1][x2,y2]$ puede implicar la visita de nodos tanto a la izquierda como a la derecha del nodo raíz.

[V] Un árbol B de orden 5 y altura 3 puede indexar 100 datos.

Pregunta 2:

Indicar de que manera evolucionaría un heap de enteros (implementado sobre un vector) durante la ejecución de las siguientes operaciones: $\text{push}(6)$, $\text{push}(2)$, $\text{push}(4)$, $\text{push}(10)$, $\text{push}(1)$, $\text{push}(15)$, $\text{push}(14)$, $\text{pop}()$, $\text{push}(3)$, $\text{push}(5)$, $\text{push}(8)$, $\text{pop}()$, $\text{pop}()$. Mostrar el estado de la estructura de datos cada 3 operaciones.

Pregunta 3:

(a) Implementad la operación $\text{Matriz} \langle T \rangle \text{ Matriz} \langle T \rangle :: \text{operator} + (\text{const Matriz} \langle T \rangle \&a)$; que se encarga de sumar dos matrices considerando que el patrón $\text{Matriz} \langle T \rangle$ tiene como atributos: $T \text{ **mat}$; $\text{int } n, m$;

Se recuerda que $c_{ij} = a_{ij} + b_{ij}$, $\forall i \in [0, \dots, n-1]$, $\forall j \in [0, \dots, m-1]$

(b) Implementad una función que recorra un árbol binario (AVL o ABB) por niveles.

Pregunta 4:

El siguiente diagrama representa la red WhatsApp para intercambio de mensajes entre móviles. Consta de un cliente que se instala en el móvil (representado por la clase `ClienteWhatsApp`) y un servidor (`ServidorWhatsApp`). El cliente del móvil tiene acceso a la lista de contactos del usuario en el móvil, con los que el usuario puede establecer conversaciones.

El funcionamiento del sistema es muy sencillo. Cuando el usuario selecciona un contacto y escribe un mensaje se invoca a la operación `escribirMensaje()` del cliente, que muestra el mensaje en pantalla de la siguiente forma:

Yo: <mensaje>

Y a continuación pasa toda la información al servidor central de WhatsApp (`procesarMensaje()`). Si el servidor comprueba que el destinatario no está en la lista de clientes (ya sea por no haber instalado el software de WhatsApp o por no estar online) se lanza una excepción `ExcepciónDestinatarioInexistente()`. En caso de que se localice el cliente, se llama a la operación `recibirMensaje()` del cliente WhatsApp del destinatario, que muestra el mensaje de esta manera:

<nombre emisor>: <mensaje>

Si el emisor no estuviera en la lista de contactos, se mostraría su número de teléfono.

Implementar las tres clases del diagrama y las asociaciones utilizando las estructuras de datos más adecuadas.

pkg WhatsApp

