Test-Arboles.pdf

putolvan

Estructuras de Datos no Lineales

2º Grado en Ingeniería Informática

Escuela Superior de Ingeniería
Universidad de Cádiz

- **En un subárbol de un árbol cualquiera se cumple la siguiente propiedad: en cada nodo, la suma de su altura y profundidad es constante y coincide con la profundidad de la hoja más profunda.**
  Falso. Salvo que todas las hojas estén igual de profunda, no te tienen porque coincidir con la profundidad de la hoja más profunda. Ejemplo: si el árbol tiene profundidad 7, y hay otra rama con profundidad 3, y cojo un nodo de los que estén en medio, pues la suma de la altura de ese nodo y la profundidad de ese nodo va a sumar como máximo 3, y no 7.

- **La profundidad del nodo más profundo del árbol es la altura de dicho árbol menos 1.**
  Falso. La profundidad del nodo más profundo es igual a la altura del árbol, el -1 sobra.

- **La profundidad del nodo más profundo del árbol es la altura de dicho árbol.**
  Verdadero

Arboles Binarios

- **Sea un árbol binario representado mediante una estructura enlazada de nodos. La destrucción del árbol binario debe eliminar los nodos uno a uno en postorden.**
  Verdadero

- **En la implementación vectorial del TAD árbol binario, que tiene como invariante que se colocan todos los elementos al principio del vector, el borrado es de coste O(1), pero la inserción es de coste O(n).**
  Falso

- **El recorrido en inorden de un árbol binario determina unívocamente el árbol del que procede.**
  Falso

- **Las operaciones del TAD Árbol Binario permiten insertar y eliminar hojas en nodos internos.**
  Verdadero.

- **La eficiencia espacial de la representación de un árbol binario mediante un vector de posiciones relativas será mayor cuantos más nodos falten en el nivel inferior.**
  Falso, será mayor cuantos menos nodos falten, y si faltan, que sean en los últimos niveles (Esta puesto en las diapositivas de ab posiciones relativas)

- **Sea un árbol binario implementado mediante una representación vectorial. La destrucción de un árbol binario debe eliminar todos los nodos uno a uno en postorden.**
  Falso. En cualquier representación vectorial, haces un delete[] del vector. No eliminas los nodos uno a uno.

- **En la implementación vectorial de árbol binario, que tiene como invariante que se colocan todos los elementos al principio del vector, la inserción es de coste O(1), pero el borrado de coste O(n).**
  Falso, el borrado también es de orden constante, puesto que lo que se hace es sustituir elemento a eliminar por el que está en la última posición, y se decrementa en uno la posición

fin.

Arboles AVL

- **Si implementáramos un TAD Conjunto mediante un AVL, podríamos garantizar que la operación de pertenencia de un elemento a un conjunto sea de coste logarítmico.**
  Verdadero

- **El equilibrio perfecto de un AVL es una condición demasiado fuerte y se permite un factor de equilibrio 1 o -1 en todos los nodos.**
  Verdadero

- **Es cierto que todos los AVL son ABB aunque algunos ABB no sean AVL.**
  Verdadero

- **El recorrido en inorden de un AVL proporciona el acceso en orden a sus elementos.**
  Verdadero

- **El calculo de la altura de un árbol es de O(n), excepto para los AVL, en cuyo caso el orden del calculo de la altura es logarítmico.**
  Falso

- **Un AVL es un ABB y el recíproco es cierto.**
  Falso. Todos los ABB no son AVL, puesto que, al no tener la propiedad de auto equilibrado, el árbol resultante no tiene por qué estar los más equilibrado posible como pasa en los AVL.

- **Es cierto que todos los AVL son ABB, pero no es cierto que algunos ABB no sean AVl.**
  Falso, (hay las dobles negaciones, que pillines hijos de puta) es cierto que algunos ABB no son AVL, por eso, lo contrario es falso.

- **La propiedad de equilibrio de un AVL permite encontrar un elemento en un tiempo de O(log n) en el peor caso.**
  Verdadero, las búsquedas en arboles de búsquedas equilibrado, son de orden logarítmico, ya que solo recorres una rama.

- **En un AVL, cada nodo tiene un factor de equilibrio de 1, 0, o -1, y ello significa que todas las hojas se van a encontrar en el último nivel o en el penúltimo.**
  Falso, significa que, en todo momento para todos los nodos, la altura de la rama izquierda no difiera en más de una unidad de la altura de la rama derecha o viceversa.

- **La propiedad de equilibrio de un AVL no implica que su altura sea la mínima posible.**
  Verdadero, la propiedad de equilibrio de un AVL implica que la altura del subárbol izquierdo no difiere en más de una unidad a la altura del subárbol derecho, de lo que se deduce que la altura puede ser uno más del mínimo.

- **El cálculo de la altura de un árbol es de O(n), excepto para los AVL, en cuyo caso, el orden del cálculo de la altura es logarítmico.**
  Falso. Para calcular la altura hay que recorrer todos los nodos O(n).

- **Una cola con prioridad representada mediante un APO permite eliminar el elemento**

prioritario con un coste O(1) en el caso peor.

Falso, eliminar un nodo es de orden logarítmico, porque hay que hundir el nodo que cambias por el de la cima, recorriendo una rama del árbol para colocarlo en su posición correcta según el criterio de ordenación que sigue el APO.

Arboles Generales

- **Se define el desequilibrio de un árbol general como la máxima diferencia entre las alturas de los subárboles mas bajo y mas alto de cada nivel. Esta definición y la diferencia de longitudes entre la rama mas larga y mas corta de dicho árbol son equivalentes.**

  Falso

- **La representación del TAD Árbol general mediante lista de hijos, es mas eficiente en espacio cuanto mas bajo es el árbol para un número determinado de nodos.**

  Falso

- **A partir de los recorridos en postorden e inorden de un árbol general, es posible reconstruir el árbol original, si, además, se conoce el grado del árbol.**

  Falso. No puedes reconstruir un árbol general a partir de ningún recorrido, aunque conozcas el grado, puesto que el grado del árbol no coinciden con el grado de todos los nodos.

- **La representación del TAD Árbol general mediante lista de hijos es más ineficiente cuanto más alto es el árbol.**

  Falso, será más ineficiente cuanto más ancho sea, puesto que habrá más hermanos derechos, y las listas de hijos serán más largas, por lo que las búsquedas en ellas también.

Arbol B

- **En un árbol B de orden m, todos los nodos contienen un mínimo de m-1/2 parte entera por defecto claves y un máximo de m-1.**

  En un árbol B de orden m, todos los nodos contienen un mínimo de $\lfloor \frac{m-1}{2} \rfloor$ claves, y un máximo de $m-1$.

  Falso

- **Al insertar en un árbol B, si el nuevo elemento no cabe en el nodo que le correspondería, se divide el nodo en dos y se promociona un elemento al nodo padre, si dicho padre no existe, se crea una nueva raíz, y en ese caso se permite que contenga un solo elemento, independientemente del mínimo permitido según el orden del árbol.**

  Verdadero

- **Para conseguir que la anchura de un árbol B sea menor, me interesa crear nodos con el mayor tamaño posible.**

  Falso

- **En el algoritmo de eliminación de una clave en un árbol B, si dicha clave no está en una hoja, se sustituye por su sucesor más inmediato, que se encontrara en el hijo derecho del nodo que contiene la clave a eliminar.**

  Falso, si el elemento que quieres eliminar esta en el nodo raíz, se sustituye por el menor de los mayores, que se encuentra en el hijo izquierdo, y no necesariamente es hijo del nodo que

contiene la clave a eliminar. (Mirar el ejemplo de clase cuando elimina la clave 20)

- **Al insertar un Árbol B, si el nuevo elemento no cabe en el nodo que le correspondería, se divide en dos y se promociona un elemento al nodo padre, y en ese caso se permite que exista algún nodo con un solo elemento, independientemente del mínimo permitido según el orden del árbol.**
  Verdadero, el nodo padre es el único al que se le permite tener un número de elementos inferior al mínimo.

APO

- **La altura de un APO depende del orden de inserción de sus elementos.**
  Falso

- **No es verdad que todos los APO tengan un nivel de desequilibrio menor o igual que 1 (en valor absoluto).**
  Falso

- **La propiedad de orden parcial de un APO implica que siempre va a estar equilibrado.**
  Falso

- **Todo APO min-max cumple estrictamente las condiciones que hemos definido para un APO, y el reciproco no es cierto.**
  Falso

- **El TAD APO permite eliminar cualquier elemento con un coste en O(log n)**
  Falso

- **Si un árbol no es un APO, tiene un desequilibrio en valor absoluto mayor o igual que 1, pero el reciproco no es cierto.**
  Falso

- **La altura mínima de un APO es logaritmo en base 2 de n por defecto, siendo n el número de elementos.**
  La altura mínima de un APO es $\lfloor \log_2 n \rfloor$, siendo n el número de elementos

  Verdadero

- **La representación vectorial de posiciones relativas es adecuada solamente para arboles parcialmente ordenados.**
  Falso, para arboles binario también.

- **Los elementos de un APO se obtienen en orden mediante la extracción sucesivas de estos.**
  Verdadero, puesto que, en la cima, siempre va a estar el elemento mínimo, y de ahí es de donde extraemos.

- **Todo APO min-max cumple estrictamente las condiciones que hemos definido para un APO, pero el reciproco no es cierto.**
  Falso. En un APO, cada nodo es menor que toda su descendencia y que el árbol es completo. En un APO min-max, cada nodo no es menor que su descendencia. Por tanto, un

APO min-max no cumple las condiciones de un APO, y el reciproco menos aún.

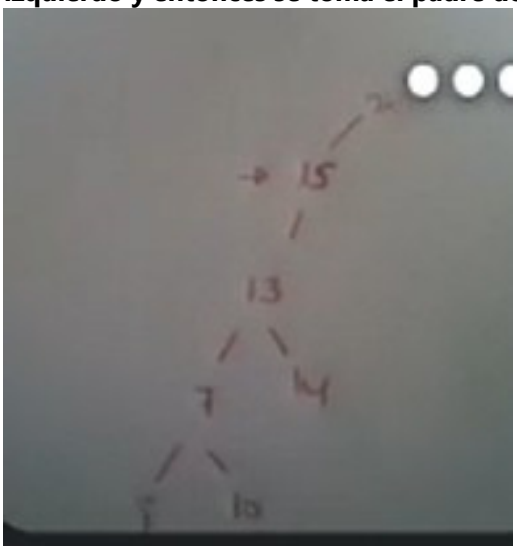
- **La propiedad de orden parcial de un APO implica que siempre va a estar equilibrado.**
  Falso, la propiedad de orden de un APO permite efectuar las inserciones y eliminaciones de nodos en un tiempo O(h) en el peor caso, siendo h la altura, que será la parte entera por defecto del log n (La altura viene dada por la propiedad de completitud), lo que significa, que las inserciones y eliminaciones en un APO son O(log n).

ABB

- **Cualquier recorrido en profundidad de un ABB proporciona el acceso en orden a sus elementos.**
  Falso

- **Si implementamos un TAD Conjunto mediante un ABB, no podríamos garantizar que la operación de pertenencia de un elemento a un conjunto sea de coste logarítmico.**
  Verdadero

- **En un ABB, sea x un elemento sin subárbol derecho. Si existe el sucesor de x, se encuentra ascendiendo hacia la raíz hasta encontrar un nodo (que puede ser el propio x) que sea hijo izquierdo y entonces se toma el padre de este.**



  Verdadero

- **Si implementamos un TAD Conjunto mediante un ABB, podríamos garantizar que la operación de pertenencia de un elemento a un conjunto siempre sea de coste logarítmico.**
  Falso. Una búsqueda en un ABB no es siempre de orden logarítmico, en el peor caso sería de orden n.

- **La inserción en el mismo orden de un conjunto de elementos en un ABB y un AVL daría como resultado arboles de la misma altura.**

  Falso. Si fuera en dos ABB (o dos AVL) si darían como resultado el mismo árbol, con la misma altura. Pero los AVL tienen la propiedad de auto equilibrado, por lo que la estructura del árbol se va a ir modificando con cada inserción, para que el árbol resultante este lo más equilibrado posible, en los ABB esto no ocurre, por lo que los árboles resultantes serian diferentes, y la altura no tendría por qué ser la misma.

- **Supongamos un ABB con el elemento x en una hoja cuyo padre tiene el valor y. Entonces, y es el menor elemento mayor que x o bien, y es el mayor elemento menor que x.**
  Verdadero. Es el mayor de los menores o el menor de los mayores. Es lo que se hace en el suprimir.