

1.- Al realizar el recorrido de un grafo es necesario ir marcando los nodos visitados, excepto si el grafo es conexo y acíclico

Verdadero

2.- En el TAD Partición, en la representación de bosques de árboles, no conviene utilizar al mismo tiempo la unión por altura y la compresión de caminos

Falsa

3.- Las operaciones del TAD árbol binario permiten insertar y eliminar hojas y nodos internos

Falso

4.- 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, y en este caso, se permite que exista algún nodo con un solo elemento, independientemente del mínimo permitido según el orden del árbol.

VERDADERO

5.- 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

6.- No ha sido fácil, pero en la última representación del TAD Partición, por fin hemos conseguido al mismo tiempo garantizar que tanto la operación de unión como la de encontrar estén en $O(1)$.

Falso

7.- 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

8.- Es cierto que todos los AVL son ABB, pero no es cierto que algunos ABB no sean AVL

Falso

9.- No es necesario que Kruskal verifique que no se producen ciclos en la solución, pues al pertenecer los nodos unidos por la arista seleccionada al mismo árbol, ello simplemente no es posible

FALSO

10.- Si no pusiéramos la función suma en los algoritmos de Floyd o Dijkstra, en el fondo no pasaría nada, pero aporta legibilidad al código y eso es muy importante

Falso, es necesario para la suma de los dos parámetros (grafo y vector) ya que si uno de los dos es INFINITO la suma es INFINITO ⇒ REVISADA

11.- Sea un árbol binario implementado mediante una representación vectorial. La destrucción del árbol binario debe eliminar los nodos uno a uno es postorden

Falso

12.- Siempre que tengamos un origen o un destino definido, debemos usar Dijkstra o Dijkstra inverso en vez de Floyd, por cuestiones de eficiencia, a la hora de calcular el árbol de expansión mínimo.

Falso

13.- 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 encontrará siempre en el hijo derecho del nodo que contiene la clave a eliminar

FALSO

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

Falso el AVL tiene función de auto equilibrio el ABB no

15.- Un recorrido en anchura desde un vértice origen permite encontrar el conjunto de vértices de un componente conexo

VERDADERO

16.- 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

17.- 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 ⇒ REVISADA

18.- Los elementos de un APO se obtienen en orden mediante la extracción sucesiva de estos.

Verdadero ⇒ REVISADA

19.- En Prim, en caso de empate entre dos aristas candidatas, es decir, del mismo coste y ambas válidas, debe escogerse la que consiga unir más nodos

Falso, coge la arista que ya esté en el árbol de expansión ⇒ REVISADO

20.- Ni Dijkstra ni Floyd funcionan correctamente con costes negativos, al contrario que Prim y Kruskal.

Falsa

21.- Prim y Kruskal no devuelven el mismo resultado siempre, excepto en el caso que haya empates entre diferentes árboles generadores mínimos.

Falso

22.- La propiedad de equilibrio de un AVL permite encontrar un elemento en un tiempo de $O(\log n)$ en el caso peor

VERDADERO

23.- La representación vectorial de posiciones relativas es adecuada solamente para árboles parcialmente ordenados

Falso ⇒ REVISADA

Para árboles completos

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

Verdadero ⇒ REVISADA

25.- 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

26.- Un AVL es ABB y el recíproco es cierto.

FALSO

27.- La propiedad de equilibrio de un AVL no implica que su altura sea la mínima posible

Verdadero

28.- En la representación por bosque de árboles del TAD Partición, dado que la función encontrar lo que recorre es el camino hasta la raíz, es en el caso peor de orden logarítmico

Falso

29.- Todo APO min-max cumple estrictamente con las condiciones que hemos definido para un APO, pero el recíproco no es cierto

falso

30.- 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

31.- Es necesario marcar los nodos visitados en el recorrido de un grafo para evitar provocar ciclos en el mismo

Falso

32.- Árbol de expansión de coste mínimo solo hay uno, eso sí, arboles de expansión en general (que no sean de coste mínimo), por supuesto hay muchos

Falso

33.- Se podría optimizar el algoritmo de Dijkstra usando un APO, en particular a la hora de buscar el nodo más cercano al origen que todavía no ha sido usado para mejorar a los demás

Verdadero

34.- 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

35.- En un grafo no dirigido, los resultados devueltos por Dijkstra y Dijkstra inverso aplicados al mismo nodo como origen y destino son iguales

Verdadero

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

FALSO

37.- Una cola con prioridad mediante un APO permite eliminar el elemento prioritario con un coste $O(1)$ en el caso peor

Falso

38.- La profundidad del nodo más profundo del árbol es la altura de dicho árbol menos 1

Falso

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

FALSO

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

Falso

Preguntas Septiembre 2020 (Corregidas por Mayte)

1. Es un árbol APO si tiene un desequilibrio en valor absoluto menor o igual que 1 pero el recíproco no es cierto.
 - **Verdadero** Aquí el recíproco quiere decir de que si el desequilibrio fuera menor o igual que es 1 es un árbol APO no es falso por lo tanto el recíproco es verdadero
2. Ni dijkstra ni floyd funcionan correctamente con valores negativos al contrario que Prim y Kruskal.
 - **Verdadera.** porque Prim y Kruskall no nos dicen nada de los costes negativos y con Dijkstra y Floyd funcionan pero no correctamente.
3. En un AVL cada nodo tiene un factor de equilibrio de 1,0,-1 y ello significa que todas las hojas se van a encontrar en el último nivel o en el penúltimo.
 - **False.** Transparencias de los avl salen hay hojas más para arriba no necesariamente tiene que estar en estos niveles cada nodo tiene que cumplir eso pero no tienen porque estar en el penúltimo o último nivel. Lo de la pregunta es lo que pasaría en los **COMPLETOS**.
4. La eficiencia espacial de la representación de un árbol binario mediante un vector de posiciones relativas será mayor cuantos menos nodos falten en el nivel inferior.
 - **Verdadero.** Porque como es un árbol completo tu guardas espacio para todos. Cuantos menos falten más completos están
5. Sea un árbol binario implementado mediante una representación vectorial. La destrucción del árbol binario No debe eliminar los nodos uno a uno en postorden.
 - **Verdadero.** Hombre si es vectorial es un vector eliminó el vector.
6. Prim y Kruskal devuelven el mismo resultado excepto posiblemente en caso de empate entre diferentes árboles generadores de coste mínimo.
 - **Verdadero.** Al haber distintos árboles generadores de coste mínimo puede dar uno u otro. El resultado es el mismo pero no tiene porque ser el mismo árbol.

7. Hemos utilizado un APO en la implementación de Prim y Kruskal para seleccionar las aristas candidatas de menor coste en orden $O(1)$
- **Falso.** En coste 1 no, tú usas el APO para seleccionar las aristas candidatas pero el coste no es $O(1)$ por que la tienes que quitar del APO y eso tiene orden $\log n$ no orden 1.
8. En un árbol B de orden m TODOS los nodos contienen un mínimo de $(m-1/2)$ claves y un máximo de m-1
- **Falso.** Todos no porque en la definición hay una excepción que es el **NODO RAÍZ**.
9. Sea X un elemento del ABB el sucesor de x se encuentra ascendiendo hacia la raíz hasta encontrar un nodo (que puede ser el propio x) con hijo derecho y a continuación buscando el menor elemento en el subárbol derecho.
- **FALSO.** Te está diciendo que tienes un nodo en el ABB y te dice que el sucesor de ese nodo está en su subárbol derecho el menor de los derechos pero esto es mentira porque si no tengo derecho su sucesor es el padre y no lo indica en el enunciado.
10. Todo APO MIN-MAX NO cumple estrictamente las condiciones que hemos definido para APO el recíproco también es cierto.
- **Verdadero.** Porque un apo minimax no es un Apo y un Apo no es un Apo min max entonces es verdad.
11. Al insertar en un árbol B, si el nuevo elemento no cabe en el nodo que le correspondiera, se divide el nodo en dos y se promociona la mediana al nodo padre, y en el caso de que el árbol conserve su altura, se permite que existan nodos con menos elementos de la mitad(por defecto) de la capacidad del nodo.
- **Falso.** Dice que la altura no aumenta un nivel. Porque eso solamente pasa cuando hay una nueva raíz que es el único nodo que puede tener menos de la mitad.
12. Se define el desequilibrio de un árbol general como la máxima diferencia entre las alturas de los subárboles más alto y más bajo de cada nivel. Esta definición y la diferencia de las longitudes entre la rama más larga y más corta de dicho árbol son equivalentes.
- **Falso.** Discusión en el foro
13. Es muy importante escoger un valor de infinito que esté fuera del rango de valores de las operaciones aritméticas con costes que va a hacer, por ejemplo, el algoritmo de Floyd o de Dijkstra en caso contrario no sirve
- **Verdadero.** El valor que tú le das a la constante infinita tiene que ser un valor que no sea ninguno de los que te vayan a salir en las operaciones que hagas con los costes.
14. La extracción sucesiva de los elementos de un APO no tiene porque devolverlos en orden.
- **Falso.** Lo devuelve en orden. Va sacando siempre el menor.
15. La propiedad de búsqueda de un ABB permite encontrar un elemento en un tiempo $O(n)$ en el peor caso.
- **Verdadero**
16. La condición de equilibrio imperfecto de un AVL NO asegura que la inserción de un elemento se pueda hacer con un coste logarítmico en el peor caso.
- **Falso**

17. Es necesario marcar los nodos visitados en el recorrido de un grafo para evitar provocar ciclos en el mismo.

- **Falso.** Yo no puedo provocar ciclos en un grafo, o es cíclico o no lo es, lo que evitamos es procesarlo más de una vez.

18. La propiedad de completitud de un apo implica que siempre va estar equilibrado.

- **Falso,** implica que las op de inserción y eliminación son $O(\log n)$

19. Es cierto que todos los AVL son ABB pero el recíproco no.

- **Verdadero.** Todos los AVL SON ABB pero al revés no. un ABB puede ser avl si está equilibrado.

20. Un APO permite inserciones en coste $O(\log n)$ en el peor de los casos

- **Verdadero.**

21. Si no pudiéramos la función suma en los algoritmos de Floyd o Dijkstra, en el fondo no pasaría nada, pero aporta legibilidad al código y eso es muy importante

- **Falso,** es necesario para la suma de los dos parámetros (grafo y vector) ya que si uno de los dos es INFINITO la suma es INFINITO.

22. La propiedad de búsqueda de un ABB permite encontrar un elemento en un tiempo $O(\log n)$ en el peor caso.

- **FALSO.** Al ser ABB consiguen que las operaciones de búsqueda y de inserción sean para el caso promedio $O(\log n)$ pero en el peor caso que el árbol haya degenerado en una lista sería $O(n)$.
Si el árbol no está balanceado, es orden lineal.

23. No aporta nada la utilización de la compresión de caminos cuando ya estamos usando la unión por altura, dado que al no actualizar la altura, no es posible combinar adecuadamente ambas técnicas.

- **Falso.** Puedes usar ambas técnicas a la vez para asegurarte en el peor caso el $\log n$ y después de muchas ejecuciones podría llegar a órdenes cercanos a la unidad.
- Se pueden combinar ambas técnicas, para quedar cercano a orden constante

24. La eliminación de un ABB puede llegar a tener coste $O(n)$.

- **Verdadero.** Si este está degenerado en una lista.
- Si está totalmente desequilibrado, puede tenerlo

25. En Prim, en caso de empate entre dos aristas candidatas, es decir, del mismo coste y ambas válidas, debe escogerse la que consiga unir más nodos

- **Falso.** coge la arista que ya esté en el árbol de expansión.

26. El recorrido en preorden de un ABB determina unívocamente el ABB del que procede.

- **Verdadera.** Si la lista no está vacía el elemento que se encuentra más a la izquierda será el nodo raíz. Al ser ABB los elementos que se encuentren en el hijo izquierdo serán menores por lo tanto recorremos de forma recursiva los hijos izquierdos y luego pasamos a los derechos. puede ser que cuente el caso de que la lista esté vacía y sea falsa?¿

27. Árbol de expansión de coste mínimo solo hay uno, eso sí, árboles de expansión en general (que no sean de coste mínimo), por supuesto hay muchos. (32 junio)

- **Falso,** hay muchos de coste mínimo REVISADA

28. En un grafo no dirigido, los resultados devueltos por Dijkstra y Dijkstra inverso aplicados al mismo nodo como origen y destino son iguales.
- Verdadero.
29. La representación del TAD Árbol general mediante listas de hijos es más ineficiente cuanto más alto es el árbol.(40 junio)
- Falso.
 - No tiene nada que ver con la altura, sino con la anchura
30. Los nuevos elementos de un árbol B se insertan en las hojas y si es necesario se reorganiza el árbol.
- Verdadero para mí.
 - Insertas por abajo del todo y crece para arriba.
31. No es necesario que Kruskal verifique que no se producen ciclos en la solución, pues al pertenecer los nodos unidos por la arista seleccionada al mismo árbol, ello simplemente no es posible. (9 Junio)
- FALSO
32. Un árbol completo se puede almacenar muy eficientemente en un vector de posiciones relativas.
- Verdadero. Ya que reservamos memoria para todo el árbol al tratarse de un vector de posiciones relativas por lo tanto al ser completo no se desperdiciará espacio de almacenamiento.
33. Al realizar el recorrido de un grafo es necesario ir marcando los nodos visitados, excepto si el grafo es conexo y acíclico (1 junio)
- Verdadero \Rightarrow REVISADA al ser acíclico no se producen ciclos y al ser conexo existe un camino entre A y B.
34. El parámetro de entrada/salida de Floyd es una matriz de camino.
- Verdadera.
 - Parámetro de E/S es el camino, se refiere al que va por referencia
35. La inserción en el mismo orden de un conjunto de elementos en un ABB y un AVL daría como resultado árboles de la misma altura (14 junio)
- Falso el AVL tiene función de auto equilibrio el ABB no \Rightarrow REVISADA
36. En un subárbol de un árbol cualquiera, tal que todas las hojas del subárbol están en el mismo nivel, se cumple la siguiente propiedad: en cada nodo la suma de su altura y su profundidad es constante y coincide con la profundidad de las hojas.
- Verdadero.
 - Si todas las hojas están al mismo nivel entonces sí.
37. Dados los recorridos en preorden y postorden de un árbol binario se puede reconstruir de forma unívoca el árbol original.
- Falso. Necesitamos el recorrido en inorden para saber los subárboles.
38. Una cola con prioridad mediante un APO permite eliminar el elemento prioritario con un coste $O(\log n)$ en el caso peor
- Verdadero
39. El recorrido en anchura de un APO proporciona el acceso en orden a los elementos almacenados.
- Falso.
40. La función ContarNodos es normalmente de coste n , excepto que sepamos que el árbol tiene como máximo en valor absoluto un desequilibrio de 1, en cuyo caso su orden es logarítmico.

- **Verdad.** Pq si fuese AVL-APO tienen un desequilibrio de 1 y si sería logarítmico.