

PREGUNTAS TIPO TEST EDNL

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 boques de árboles, no conviene utilizar al mismo tiempo la unión por altura y la compresión de caminos.
FALSO
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.
FALSO
5. Si implementáramos 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
11. Sea un árbol binario implementado mediante una representación vectorial. La destrucción del árbol binario debe eliminar los nodos uno a uno en postorden.
VERDADERO
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 arboles de la misma altura.
FALSO
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
18. Los elementos de un APO se obtienen en orden mediante la extracción sucesiva de estos.
VERDADERO
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
20. Ni Dijkstra ni Floyd funcionan correctamente con costes negativos, al contrario que Prim y Kruskall.
VERDADERO
21. Prim y Kruskall 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 arboles parcialmente ordenados.
FALSO
24. Para conseguir que la anchura del árbol B sea menor, me interesa crear nodos con el mayor tamaño posible.
VERDADERO
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.
VERDADERO
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.
FALSO
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.
VERDADERO

37. La profundidad del nodo más profundo del árbol es la altura de dicho árbol menos 1.
FALSO
38. 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
39. La representación del TAD Árbol general mediante listas de hijos es más ineficiente cuanto más alto es el árbol.
FALSO
40. La propiedad de búsqueda de un ABB permite encontrar un elemento en un tiempo de $O(\log n)$ en el caso peor.
FALSO
41. 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
42. La eliminación de elemento en un ABB puede llegar a tener un coste $O(n)$.
VERDADERO
43. El recorrido en preorden de un ABB determina unívocamente el ABB del que procede.
VERDADERO
44. 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 el de Dijkstra. En caso contrario no sirve.
VERDADERO
45. Es cierto que todos los AVL son ABB, pero el recíproco no es cierto.
VERDADERO
46. Hemos utilizado un APO en la implementación de Prim y Kruskall para seleccionar la arista candidata de menor coste en un tiempo $O(1)$.
FALSO
47. Sea x un elemento de un 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
48. En un grafo no dirigido, los resultados devueltos por Dijkstra y Dijkstra inverso aplicados al mismo nodo como origen y destino son iguales.
VERDADERO
49. Los nuevos elementos en un árbol B se insertan en las hojas y, si es necesario, se reorganiza el árbol.
VERDADERO
50. En un árbol B de orden m , todos los nodos contienen un mínimo de $\left\lfloor \frac{m-1}{2} \right\rfloor$ claves, y un máximo de $m-1$.
FALSO
51. La condición de equilibrio no perfecto de un AVL no asegura que la inserción de un elemento se pueda hacer a un coste de $O(\log n)$ en el peor caso.
FALSA
52. Es necesario que Kruskall verifique que no se producen ciclos en la solución, lo cual queda garantizado al seleccionar la arista de menor coste cuyos nodos pertenecen a diferentes árboles.
VERDADERO
53. Se define el desequilibrio de un árbol general como la máxima diferencia entre las alturas de los subárboles más bajo y más alto de cada nivel. Esta definición y la diferencia de longitudes entre la rama más larga y más corta de dicho árbol son equivalentes.
VERDADERO
54. Un árbol completo se puede almacenar muy eficientemente en un vector de posiciones relativas.
VERDADERO
55. El parámetro de entrada/salida de Floyd es una matriz de caminos.
VERDADERO

56. Prim y Kruskall devuelven el mismo resultado, excepto posiblemente en el caso que haya empates entre diferentes árboles generadores mínimos.
FALSO
57. Si un árbol es un APO, tiene un desequilibrio en valor absoluto menor o igual que 1, pero el recíproco no es cierto.
VERDADERO
58. 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 profundidad es constante y coincide con la profundidad de las hojas.
VERDADERO
59. Dados los recorridos en preorden y postorden de un árbol binario se puede reconstruir unívocamente el árbol original.
FALSO
60. 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 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 de un nodo.
FALSO
61. La propiedad de completitud de un APO implica que siempre va a estar equilibrado.
VERDADERO
62. Una cola de prioridad representada mediante un APO permite inserciones con un coste en $O(\log n)$ en el caso peor.
VERDADERO
63. El recorrido en anchura de un APO proporciona el acceso en orden a los elementos almacenados.
FALSO
64. La función ContarNodos es normalmente de coste n , excepto que sepamos que el árbol binario tiene como máximo en valor absoluto un desequilibrio de 1, en cuyo caso su orden es logarítmico.
FALSO
65. Todo APO min-max cumple estrictamente con las condiciones que hemos definido para un APO, y el recíproco también es cierto.
FALSO
66. La extracción sucesiva de los elementos de un APO no tiene por qué devolverlos en orden.
FALSO
67. 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
68. 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
69. En el TAD partición, en la representación de bosque de árboles, no conviene utilizar al mismo tiempo la unión por altura y la compresión de caminos.
FALSO
70. 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
71. 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
72. 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.
FALSO
73. Un AVL es un ABB y el recíproco es cierto.
FALSO

74. 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
75. Una cola con prioridad representada mediante un APO permite eliminar el elemento prioritario con un coste $O(1)$ en el caso peor.
FALSO
76. 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 ese caso se permite que exista algún nodo con un solo elemento, independientemente del mínimo permitido según el orden del árbol.
FALSO
77. La propiedad de equilibrio de un AVL no implica que su altura sea la mínima posible.
VERDADERO
78. Sea un árbol binario implementado mediante una representación vectorial, la destrucción del árbol binario debe de eliminar los nodos uno a uno en postorden.
VERDADERO
79. Una cola con prioridad representada mediante un APO permite eliminar el elemento prioritario con un coste $O(\log n)$ en el caso peor.
VERDADERO