

1. ¿Qué es un árbol binario?
  - a. a) Un árbol en el que cada nodo tiene un número ilimitado de hijos.
  - b. b) Un árbol en el que cada nodo tiene dos hijos o ninguno.
  - c. c) Un árbol en el que cada nodo tiene un hijo.
2. ¿Qué es un árbol AVL?
  - a. a) Un árbol binario en el que la altura de las dos subárboles de cualquier nodo difiere en uno o menos.
  - b. b) Un árbol binario en el que la altura de las dos subárboles de cualquier nodo difiere en dos o menos.
  - c. c) Un árbol binario en el que la altura de las dos subárboles de cualquier nodo difiere en tres o menos.
3. ¿Qué es un recorrido DFS?
  - a. a) Un recorrido en el que se visitan los nodos de un árbol desde la raíz hasta las hojas.
  - b. b) Un recorrido en el que se visitan los nodos de un árbol desde las hojas hasta la raíz.
  - c. c) Un recorrido en el que se visitan los nodos de un árbol de forma profundidad primero.
4. ¿Qué es un recorrido BFS?
  - a. a) Un recorrido en el que se visitan los nodos de un árbol desde la raíz hasta las hojas.
  - b. b) Un recorrido en el que se visitan los nodos de un árbol desde las hojas hasta la raíz.
  - c. c) Un recorrido en el que se visitan los nodos de un árbol de forma anchura primero.
5. ¿Qué algoritmo de ordenamiento tiene complejidad  $O(n^2)$ ?
  - a. a) Quick Sort.
  - b. b) Merge Sort.
  - c. c) Bubble Sort.
6. ¿Qué algoritmo de ordenamiento utiliza una técnica de "divide y vencerás"?
  - a. a) Selection Sort.
  - b. b) Insertion Sort.
  - c. c) Merge Sort.
7. ¿Qué algoritmo de ordenamiento utiliza una técnica de "partición"?
  - a. a) Quick Sort.
  - b. b) Bubble Sort.
  - c. c) Insertion Sort.

8. ¿Cuál de los siguientes algoritmos de ordenamiento es más eficiente?
- a. a) Bubble Sort.
  - b. b) Merge Sort.
  - c. c) Selection Sort.
9. ¿Qué algoritmo de ordenamiento es útil para ordenar listas encadenadas?
- a. a) Bubble Sort.
  - b. b) Quick Sort.
  - c. c) Insertion Sort.
10. ¿Qué algoritmo de ordenamiento utiliza una técnica de "cubo"?
- a. a) Pigeonhole Sort.
  - b. b) Radix Sort.
  - c. c) Merge Sort.
11. ¿Qué algoritmo de ordenamiento tiene complejidad  $O(n)$ ?
- a. a) Bubble Sort.
  - b. b) Merge Sort.
  - c. c) Pigeonhole Sort.
12. ¿Qué algoritmo de ordenamiento es útil para ordenar datos numéricos con muchas repeticiones?
- a. a) Quick Sort.
  - b. b) Radix Sort.
  - c. c) Merge Sort.
13. ¿Qué algoritmo de ordenamiento es adecuado para ordenar grandes cantidades de datos de forma distribuida?
- a. a) Bubble Sort.
  - b. b) Radix Sort.
  - c. c) Merge Sort.
14. ¿Cuál es la complejidad del algoritmo Quick Sort en el peor caso?
- a. a)  $O(n)$ .
  - b. b)  $O(n \log n)$ .
  - c. c)  $O(n^2)$ .
15. ¿Cuáles son algunas de las ventajas de utilizar un algoritmo de ordenamiento distributivo como el radix sort?
- a. A. Es estable y no requiere memoria adicional.
  - b. B. Es muy eficiente para ordenar grandes cantidades de datos.

- c. C. Es fácil de implementar y entender.
- d. D. No depende de la magnitud de los datos a ordenar.
- e. E. Todas las anteriores son correctas.
- f. Ninguna de las anteriores es correcta.

### Respuestas correctas

- 1. A, C
- 2. B, D
- 3. A
- 4. C, D
- 5. B
- 6. A, B, C
- 7. C
- 8. B, C
- 9. A, B
- 10. C
- 11. B, C
- 12. A, B
- 13. A, B, C
- 14. B, C
- 15. B, C