

La siguiente es una lista de preguntas de teoría sobre los temas que se incluyen en el examen parcial de ADA. En el examen se pide además práctica en la solución de problemas.

1. Tema I. Análisis de Algoritmos

1. ¿Qué entendemos por eficiencia o coste de un algoritmo?
2. ¿Qué es el tiempo de ejecución de un algoritmo?
3. ¿Y el espacio de memoria?
4. ¿Cómo se define el coste de un algoritmo en el caso mejor?
5. ¿Y en el caso peor?
6. ¿Y en el caso promedio?
7. ¿Cómo se define la clase de funciones O ?
8. ¿Qué propiedades tiene?
9. ¿Cómo se define la clase de funciones Ω ?
10. ¿Qué propiedades tiene?
11. ¿Cómo se define la clase de funciones Θ ?
12. ¿Qué propiedades tiene?
13. ¿Cómo se calcula el coste de los algoritmos iterativos?
14. ¿Y el de los algoritmos recursivos?
15. ¿Qué es una recurrencia sustractora?
16. ¿Qué dice el teorema maestro para recurrencias sustractoras?
17. ¿Qué es una recurrencia divisora?
18. ¿Qué dice el teorema maestro para recurrencias divisoras?
19. ¿Cómo funciona el método de ordenación por selección?
20. ¿Qué costes tiene en los casos mejor, peor y promedio?
21. ¿Cómo se implementa en C++?
22. ¿Cómo funciona el método de ordenación por inserción?
23. ¿Qué costes tiene en los casos mejor, peor y promedio?
24. ¿Cómo se implementa en C++?

25. ¿Qué ventajas y desventajas tienen los métodos anteriores?
26. ¿En qué casos conviene más utilizar el método de ordenación por selección en vez del de ordenación por inserción?

2. Tema II. Esquema de Dividir y Vencer

1. ¿En qué consiste el esquema general de dividir y vencer?
2. ¿Cómo funciona el algoritmo de búsqueda dicotómica (o binaria)?
3. ¿Cómo se implementa en C++?
4. ¿Qué costes tiene en los casos mejor, peor y promedio?
5. ¿Cómo funciona el algoritmo de ordenación por fusión?
6. ¿Cómo se implementa en C++?
7. ¿Qué costes tiene en los casos mejor, peor y promedio?
8. ¿Qué ventajas y qué desventajas tiene?
9. ¿Cómo funciona el algoritmo de ordenación rápida?
10. ¿Cómo se implementa en C++?
11. ¿Qué costes tiene en los casos mejor, peor y promedio?
12. ¿Qué ventajas y qué desventajas tiene?

3. Tema III. Ampliación de Estructuras de Datos

3.1. Diccionarios

1. ¿Qué es el TAD Diccionario?
2. ¿Qué es un árbol binario de búsqueda?
3. ¿Cómo se implementa en C++?
4. ¿Cómo funcionan los algoritmos para borrar, buscar e insertar elementos en árboles binarios de búsqueda?
5. ¿Cómo se implementan en C++?
6. ¿Cuáles son sus costes en el caso mejor, peor y promedio?
7. ¿Cómo funcionan las operaciones de buscar el elemento mínimo y buscar el elemento máximo en un árbol binario de búsqueda?

8. ¿Qué costes tienen en todos los casos?
9. ¿Qué es un árbol AVL?
10. ¿Cómo funciona la operación de insertar un nuevo elemento en un AVL?
11. ¿Qué costes tiene (en los casos mejor, peor y promedio)?
12. ¿Qué coste tiene en los casos mejor, peor y promedio buscar un elemento?
13. ¿Y buscar el elemento mínimo?
14. ¿Y el máximo?
15. ¿Qué es una tabla de dispersión con encadenamiento separado?
16. ¿Qué costes tienen las operaciones del TAD diccionario si éste se implementa con una tabla de dispersión con encadenamiento separado?
17. ¿Y con encadenamiento abierto?
18. ¿Y con encadenamiento abierto lineal?
19. ¿Cómo implementar en C++ el TAD diccionario utilizando tablas de dispersión?
20. ¿Qué ventajas y qué desventajas tienen las implementaciones del TAD Diccionario mencionadas anteriormente?
21. ¿Cuándo utilizar unas y otras?

3.2. Colas de Prioridad

1. ¿Qué es el TAD Cola de Prioridad?
2. ¿Qué estructuras de datos pueden utilizarse para implementarlo?
3. ¿Cuál es la idónea?
4. ¿Cómo se implementa utilizando árboles y cómo funcionan todas las operaciones?
5. ¿Y utilizando vectores?
6. ¿Qué costes tienen en el caso mejor, peor y promedio las operaciones del TAD Cola de Prioridad en un heap?
7. ¿Cómo funciona heapsort?
8. ¿Cómo se implementa en C++?
9. ¿Qué costes tiene?
10. ¿Qué ventajas y qué desventajas tiene?

4. Tema IV. Grafos

1. ¿Qué es un grafo?
2. ¿Qué es un digrafo?
3. ¿Cómo se representan grafos y digrafos utilizando matrices de adyacencia?
4. ¿Cuál es el coste en espacio de esta representación?
5. ¿Cómo se representan grafos y digrafos utilizando listas de adyacencia?
6. ¿Cuál es el coste en espacio de esta representación?
7. ¿Qué es un bucle?
8. ¿Qué es un camino?
9. ¿Qué es un ciclo?
10. ¿Qué es un grafo conexo?
11. ¿Qué es un digrafo fuertemente conexo?
12. ¿Qué es un digrafo débilmente conexo?
13. ¿Qué es el grado de un vértice en un grafo?
14. ¿Qué es el grado de un grafo?
15. ¿Qué es el grado de entrada de un vértice de un digrafo?
16. ¿Qué es el grado de salida de un vértice de un digrafo?
17. ¿Qué es el diámetro de un grafo?
18. ¿Qué es un grafo Euleriano?
19. ¿Qué es un grafo Hamiltoniano?
20. ¿Qué es un recorrido de un grafo?
21. ¿Cómo se hace el recorrido en profundidad de un grafo?
22. ¿Y el recorrido en amplitud?
23. ¿Cómo se implementan?
24. ¿Qué es la ordenación topológica de un grafo dirigido acíclico?
25. ¿Cómo se implementa?