

La siguiente es una lista de preguntas de teoría sobre los temas que se incluyen en la segunda parte del curso de ADA. Debe completarse con la primera lista (del examen parcial) para tener una guía completa de la asignatura. En los exámenes se pide además práctica en la solución de problemas.

## 1. Tema: Algoritmos de Búsqueda Exhaustiva

1. ¿En qué consiste el esquema general de vuelta atrás (backtrack)?
2. ¿Cómo se implementa este esquema sin utilizar marcajes para listar todas las posibles soluciones de un problema?
3. ¿Y utilizando marcajes?
4. ¿Y si nos basta con dar la primera solución encontrada?
5. ¿Y si se trata de un problema de optimización?
6. ¿Cuáles son los problemas tipo que se resuelven mediante esta técnica?
7. ¿En qué varía el esquema general?
8. ¿Qué pregunta intenta responder la teoría de la NP-completez?
9. ¿Porqué?
10. ¿Qué relación tiene con la búsqueda exhaustiva?
11. ¿Qué elementos conforman la versión computacional de un problema cualquiera?
12. ¿Cuál es la versión decisional de un problema computacional?
13. ¿Que hay que agregar a la formulación decisional si el problema es de optimización?
14. ¿Conoces las formulaciones decisionales de al menos tres problemas diferentes?
15. ¿Qué significa que un algoritmo sea polinómico?
16. ¿Qué ejemplos conoces de algoritmos polinómicos?
17. ¿Qué significa que un problema sea decidible en tiempo polinómico?
18. ¿Cómo se define la clase P?
19. ¿Cuál es la diferencia entre encontrar y verificar?
20. ¿Qué es un testimonio?
21. ¿Qué es un algoritmo verificador?
22. ¿Qué significa que un problema sea decidible en tiempo polinómico indeterminista?
23. ¿Cómo se define la clase NP?
24. ¿Cómo se decide si un problema pertenece a la clase NP?
25. ¿Es la clase P un subconjunto de la clase NP?
26. ¿Todo problema de la clase P también pertenece a la clase NP?

27. ¿Es la clase NP un subconjunto de la clase P?
28. ¿Todo problema de la clase NP también pertenece a la clase P?
29. ¿Qué es una reducción?
30. ¿Y una reducción polinómica?
31. ¿Qué significa que un problema sea NP-difícil?
32. ¿Y NP-completo?
33. ¿Existen problemas NP-completos?
34. ¿Conoces al menos tres problemas NP-completos "famosos"?

## **2. Tema: Programación Dinámica**

1. ¿En qué consiste el esquema general de dividir y vencer?
2. ¿Porqué en algunas ocasiones es ineficiente?
3. ¿Cómo se puede mejorar la eficiencia?
4. ¿Cuál es la metodología general de la programación dinámica?
5. ¿Cuándo tiene sentido aplicarla?
6. ¿Qué ventajas y qué desventajas tiene?
7. ¿Qué problemas sabes resolver utilizando programación dinámica?
8. ¿Qué costes tienen las soluciones en los casos mejor, peor y promedio?
9. ¿Cómo se plantean, en general, las recurrencias que resuelven el problema?
10. ¿Conoces ejemplos de problemas de optimización que se resuelvan mediante esta técnica?
11. ¿Y ejemplos de problemas de contar posibilidades?

## **3. Tema: Algoritmos Voraces**

1. ¿Qué es un algoritmo voraz?
2. ¿Qué características tiene?
3. ¿Cómo se demuestra si un algoritmo voraz es óptimo? (En el examen no se piden estas demostraciones)
4. ¿Cómo se demuestra que no lo es?
5. ¿Qué es un grafo con pesos?
6. ¿Qué es un árbol de expansión?
7. ¿Y un árbol de expansión mínimo?
8. ¿Cómo funciona el algoritmo de Prim?
9. ¿Cómo se implementa en C++?

10. ¿Qué costes tiene en todos los casos?
11. ¿Qué es un camino?
12. ¿Y un camino mínimo?
13. ¿Cómo funciona el algoritmo de Dijkstra?
14. ¿Cómo se implementa en C++?
15. ¿Qué costes tiene en todos los casos?
16. ¿Qué es el problema de la codificación de longitud mínima?
17. ¿Cómo funciona el algoritmo de Huffman?
18. ¿Cómo se implementa en C++?
19. ¿Qué costes tiene en todos los casos?