

# ADA Exam

This exam consists of 3 pages, not including this cover page. Please go through your copy to make sure that all pages are in good order. The exam consists of a set of short questions with multiple choices. There are 20 points, total, on this exam. Happy solving!

Name: \_\_\_\_\_

Date: \_\_\_\_\_

1	(A)	(B)	(C)	(D)	(E)
2	(A)	(B)	(C)	(D)	(E)
3	(A)	(B)	(C)	(D)	(E)
4	(A)	(B)	(C)	(D)	(E)
5	(A)	(B)	(C)	(D)	(E)
6	(A)	(B)	(C)	(D)	(E)
7	(A)	(B)	(C)	(D)	(E)
8	(A)	(B)	(C)	(D)	(E)
9	(A)	(B)	(C)	(D)	(E)
10	(A)	(B)	(C)	(D)	(E)
11	(A)	(B)	(C)	(D)	(E)
12	(A)	(B)	(C)	(D)	(E)
13	(A)	(B)	(C)	(D)	(E)
14	(A)	(B)	(C)	(D)	(E)
15	(A)	(B)	(C)	(D)	(E)
16	(A)	(B)	(C)	(D)	(E)
17	(A)	(B)	(C)	(D)	(E)
18	(A)	(B)	(C)	(D)	(E)
19	(A)	(B)	(C)	(D)	(E)
20	(A)	(B)	(C)	(D)	(E)



1AF9D0

**Pontificia Universidad Javeriana**  
**ADA**  
**2025 – 10**  
**3rd Exam**  
**Code: 1AF9D0**

1. ¿Cuál es una característica clave de los algoritmos de fuerza bruta?
  - A. Realizan una búsqueda parcial optimizada.
  - B. Evalúan todas las posibles soluciones sin podarlas.**
  - C. Usan estructura de divide y vencerás para mejorar el rendimiento.
  - D. Solo funcionan con entradas grandes y datos ordenados.
2. ¿Qué problema se resuelve comúnmente con fuerza bruta?
  - A. Búsqueda binaria.
  - B. Problema del Viajante (TSP).**
  - C. Árbol de expansión mínima.
  - D. QuickSort.
3. ¿Qué propiedad es esencial para aplicar programación dinámica?
  - A. Aleatoriedad en los datos.
  - B. Subproblemas superpuestos.**
  - C. Datos en forma de árbol.
  - D. Entrada ya ordenada.
4. ¿Cuál es la finalidad principal de usar memoización?
  - A. Reducir uso de memoria.
  - B. Evitar cálculos redundantes almacenando resultados.**
  - C. Generar pseudocódigo automáticamente.
  - D. Aumentar precisión numérica.
5. ¿Qué estrategia se usa en divide y vencerás?
  - A. Eliminar duplicados antes de resolver.
  - B. Dividir el problema en subproblemas, resolverlos y combinar.**
  - C. Iterar hasta que el input sea vacío.
  - D. Ordenar la entrada antes de iniciar.
6. ¿Cuál algoritmo es un ejemplo clásico de divide y vencerás?
  - A. DFS.
  - B. Fuerza bruta para TSP.
  - C. MergeSort.**
  - D. Knapsack.
7. ¿Qué propiedad garantiza que un algoritmo greedy sea óptimo?

- A. Complejidad logarítmica.
  - B. Subestructura óptima.**
  - C. Entrada ordenada.
  - D. Uso de árboles binarios.
8. ¿Cuál de los siguientes problemas se puede resolver con programación dinámica?
- A. Dijkstra.
  - B. Subset Sum.**
  - C. Búsqueda binaria.
  - D. TSP con DFS.
9. ¿Qué significa la notación  $\mathcal{O}(n^2)$ ?
- A. Cota superior del tiempo de ejecución en el peor caso.**
  - B. Cota inferior del tiempo de ejecución.
  - C. Tiempo exacto que toma el algoritmo.
  - D. Espacio de memoria utilizado.
10. ¿Qué técnica se usa para verificar la corrección de un algoritmo?
- A. Analizar el uso de memoria.
  - B. Comprobar precondiciones y postcondiciones.**
  - C. Reducir el tamaño de la entrada.
  - D. Usar programación orientada a objetos.
11. ¿Qué implica una estrategia de poda en backtracking?
- A. Eliminar caminos que no pueden llevar a soluciones válidas.**
  - B. Dividir el input en mitades iguales.
  - C. Usar estructuras de árbol AVL.
  - D. Ejecutar el algoritmo más de una vez.
12. ¿Cuál es una ventaja de usar pseudocódigo en el diseño de algoritmos?
- A. Ejecuta más rápido que el código real.
  - B. Mejora la claridad y legibilidad del algoritmo.**
  - C. Optimiza automáticamente el código.
  - D. Genera pruebas unitarias.
13. ¿Qué herramienta ayuda a identificar cuellos de botella en algoritmos?
- A. Visual Studio Code.
  - B. Perfiladores como cProfile o gprof.**
  - C. Compiladores.
  - D. Exploradores de archivos.
14. ¿Qué notación representa la cota ajustada del tiempo de ejecución?

- A.  $\Omega(n)$
  - B.  $\Theta(n)$**
  - C.  $\mathcal{O}(n)$
  - D.  $\delta(n)$
15. ¿Qué algoritmo puede beneficiarse del uso de bitmasking?
- A. MergeSort.
  - B. Solución de Sudoku por backtracking.**
  - C. BFS en árboles.
  - D. Dijkstra.
16. ¿Qué estrategia mejora la eficiencia de QuickSort?
- A. Reemplazar por fuerza bruta en todos los casos.
  - B. Usar selección aleatoria del pivote.**
  - C. Repetir partición tres veces.
  - D. Usar estructuras dinámicas.
17. ¿Qué significa una complejidad de  $\mathcal{O}(1)$ ?
- A. El algoritmo no termina nunca.
  - B. Tiempo constante, independiente del tamaño de entrada.**
  - C. Tiempo cuadrático.
  - D. Tiempo exponencial.
18. ¿Cuál es un ejemplo de estructura útil en programación dinámica?
- A. Árbol binario de búsqueda.
  - B. Tabla de memoización.**
  - C. Grafo dirigido acíclico.
  - D. Árbol AVL.
19. ¿Cuál técnica implica recorrer manualmente un algoritmo con ejemplos concretos?
- A. Refactorización.
  - B. Ejecución en seco (dry run).**
  - C. Compilación anticipada.
  - D. Optimización recursiva.
20. ¿Qué algoritmo es un ejemplo clásico de greedy?
- A. MergeSort.
  - B. Codificación de Huffman.**
  - C. Subset Sum con memoización.
  - D. Floyd-Warshall.

**End of Exam**