

Enunciados Y algoritmos

miércoles, 10 de julio de 2024 10:02 p. m.

Fácil

- ☒ 1. Suma de dos números:
 - Enunciado: Escribe un programa que lea dos números enteros del usuario y muestre su suma.
 - Input: Dos números enteros.
 - Output: La suma de los dos números.
- ☒ a. Número par o impar:
 - Enunciado: Escribe un programa que determine si un número ingresado por el usuario es par o impar.
 - Input: Un número entero.
 - Output: "Par" o "Impar".
- ☒ b. Factorial de un número:
 - Enunciado: Escribe un programa que calcule el factorial de un número ingresado por el usuario.
 - Input: Un número entero no negativo.
 - Output: El factorial del número.
- ☒ c. Conteo de vocales:
 - Enunciado: Escribe un programa que cuente el número de vocales en una cadena de texto ingresada por el usuario.
 - Input: Una cadena de texto.
 - Output: El número de vocales en la cadena.

Intermedio

- ☒ 5. Ordenar un arreglo:
 - Enunciado: Escribe un programa que ordene un arreglo de enteros usando el algoritmo de ordenamiento por burbuja (Bubble Sort).
 - Input: Un arreglo de enteros.
 - Output: El arreglo ordenado.
- ☒ 6. Número primo:
 - Enunciado: Escribe un programa que determine si un número ingresado por el usuario es primo.
 - Input: Un número entero.
 - Output: "Primo" o "No primo".
- ☒ 7. Fibonacci:
 - Enunciado: Escribe un programa que genere la secuencia de Fibonacci hasta el n-ésimo término.
 - Input: Un número entero n.
 - Output: Los primeros n términos de la secuencia de Fibonacci.
- ☐ 8. Búsqueda binaria:
 - Enunciado: Escribe un programa que implemente el algoritmo de búsqueda binaria en un arreglo ordenado.
 - Input: Un arreglo ordenado de enteros y un número entero a buscar.
 - Output: El índice del número en el arreglo o -1 si no se encuentra.

Avanzado

- ☐ 9. Camino mínimo en una matriz:
 - Enunciado: Escribe una función que resuelva el problema del camino mínimo en una matriz m x n. La función debe encontrar el camino con la suma mínima de la esquina superior izquierda a la esquina inferior derecha.
 - Input: Una matriz de enteros m x n.
 - Output: La suma mínima del camino.
- ☐ 10. Permutaciones de una cadena:
 - Enunciado: Escribe un programa que genere todas las permutaciones posibles de una cadena de texto ingresada por el usuario.
 - Input: Una cadena de texto.
 - Output: Todas las permutaciones posibles de la cadena.
- ☐ 11. Búsqueda en profundidad (DFS) en un grafo:
 - Enunciado: Implementa el algoritmo de búsqueda en profundidad (DFS) en un grafo representado por una lista de adyacencia.
 - Input: Un grafo representado por una lista de adyacencia y un nodo inicial.
 - Output: Los nodos visitados en el orden de la búsqueda en profundidad.
- ☐ 12. Problema del cambio de moneda:
 - Enunciado: Escribe un programa que, dado un conjunto de denominaciones de monedas y una cantidad total, encuentre el número mínimo de monedas necesarias para formar esa cantidad.
 - Input: Un arreglo de denominaciones de monedas y una cantidad total.
 - Output: El número mínimo de monedas necesarias para formar la cantidad total.

Expert

- ☐ 13. Problema del viajante (TSP):
 - Enunciado: Escribe un programa que resuelva el problema del viajante (TSP) usando la técnica de programación dinámica.
 - Input: Una matriz de costos que representa las distancias entre cada par de ciudades.
 - Output: La ruta de costo mínimo que visita todas las ciudades exactamente una vez y regresa a la ciudad de inicio.
- ☐ 14. Algoritmo de Dijkstra:
 - Enunciado: Implementa el algoritmo de Dijkstra para encontrar el camino más corto desde un nodo inicial a todos los demás nodos en un grafo ponderado dirigido.
 - Input: Un grafo ponderado dirigido representado por una lista de adyacencia y un nodo inicial.
 - Output: Las distancias más cortas desde el nodo inicial a todos los demás nodos.
- ☐ 15. Resolver Sudoku:
 - Enunciado: Escribe un programa que resuelva un tablero de Sudoku dado usando backtracking.
 - Input: Un tablero de Sudoku parcialmente completado.

Algoritmos de Búsqueda y Ordenamiento

- Búsqueda Lineal:** Algoritmo básico para encontrar un elemento en una lista.
- Búsqueda Binaria:** Búsqueda eficiente en listas ordenadas.
- Ordenamiento por Burbuja:** Algoritmo de ordenamiento simple.
- Ordenamiento por Inserción:** Otro algoritmo de ordenamiento simple.
- Quicksort:** Algoritmo de ordenamiento eficiente.
- Mergesort:** Algoritmo de ordenamiento basado en la técnica de divide y vencerás.
- Heapsort:** Algoritmo de ordenamiento utilizando una estructura de datos heap.

Algoritmos de Programación Dinámica

- Problema del Camino Mínimo:** Encontrar el camino mínimo en una matriz.
- Problema de la Mochila (Knapsack Problem):** Optimización de la selección de elementos para maximizar el valor total.
- Secuencia Común Más Larga (LCS):** Encontrar la subsecuencia más larga común entre dos secuencias.
- Cadena de Matrices (Matrix Chain Multiplication):** Determinar la manera más eficiente de multiplicar una cadena de matrices.

Algoritmos de Grafos

- Búsqueda en Profundidad (DFS):** Recorrido de grafos.
- Búsqueda en Amplitud (BFS):** Recorrido de grafos.
- Dijkstra:** Algoritmo para encontrar el camino más corto desde un nodo a todos los demás en un grafo con pesos no negativos.
- Floyd-Warshall:** Algoritmo para encontrar todos los caminos más cortos en un grafo.
- Prim y Kruskal:** Algoritmos para encontrar el árbol de expansión mínima (MST) en un grafo.

Algoritmos de Combinatoria y Backtracking

- Permutaciones:** Generación de todas las permutaciones de un conjunto.
- Combinaciones:** Generación de todas las combinaciones de un conjunto.
- Problema de las N-Reinas:** Colocación de reinas en un tablero de ajedrez de manera que no se ataquen entre ellas.
- Sudoku:** Resolución de puzzles de Sudoku.

Algoritmos Greedy

- Problema del Cambio de Monedas:** Selección de la menor cantidad de monedas para hacer un cambio dado.
- Actividad de Selección:** Selección del mayor número de actividades que no se superpongan.
- Huffman Coding:** Algoritmo para la compresión de datos.

Algoritmos Geométricos

- Convex Hull:** Encontrar el menor polígono convexo que contiene un conjunto de puntos.
- Segment Intersection:** Determinar si dos segmentos de línea se intersectan.

Estructuras de Datos Fundamentales

- Listas Enlazadas:** Implementación y manipulación.
- Pilas y Colas:** Estructuras básicas para almacenamiento y recuperación de datos.
- Árboles:** Árboles binarios, AVL, y árboles de búsqueda binaria.
- Grafos:** Representación y manipulación de grafos usando listas de adyacencia y matrices de adyacencia.
- Heaps:** Estructuras de datos para implementación de colas de prioridad.
- Hash Tables:** Estructuras de datos para acceso rápido a elementos.

Técnicas de Optimización y Análisis

- Notación Big-O:** Análisis de la complejidad temporal y espacial de los algoritmos.
- Divide y Vencerás:** Técnica para resolver problemas dividiéndolos en subproblemas más pequeños.
- Técnica de Ventana Deslizante:** Optimización para problemas que requieren subsecuencias o subarreglos.
- Algoritmos de Aproximación:** Soluciones cercanas a óptimas para problemas difíciles de resolver exactamente.

○ Output: El tablero de Sudoku completado.