Expo info

La programación dinámica se aplica, generalmente, para resolver problemas de optimización, como por ejemplo:

* Problemas con diversas soluciones.
* Cada solución tiene asociado un valor.
* Se busca una solución con un valor óptimo (máximo o mínimo), entre las muchas soluciones con valor óptimo que pueden encontrarse.

**Algoritmos de solución**

Para realizar el análisis de problemas que se pueden resolver a través de la técnica de la [**programación dinámica**](http://virtual.umng.edu.co/distancia/ecosistema/odin/odin_desktop.php?path=Li4vb3Zhcy9pbmdlbmllcmlhX2NpdmlsL2ludmVzdGlnYWNpb25fZGVfb3BlcmFjaW9uZXNfaWkvdW5pZGFkXzQv), existen varios algoritmos de solución.

Entre ellos se encuentran:

**La Subsecuencia Común Más Larga (LCS, por sus siglas en inglés)** es un problema clásico en informática que consiste en encontrar la secuencia más larga de elementos que aparece (en el mismo orden) en dos o más secuencias dadas. A diferencia de las subcadenas, los elementos de la subsecuencia no necesitan ser consecutivos en las secuencias originales.

Si tienes las secuencias "AGGTAB" y "GXTXAYB", la LCS es "GTAB", ya que es la secuencia más larga que aparece en ambas, aunque no de forma consecutiva.

Aplicaciones:

* [**Biología molecular**](https://www.google.com/search?client=opera-gx&sca_esv=e8b3de2c0113d5ce&cs=1&sxsrf=AE3TifNr-IS48Bl-HpCQ5YHWDlWtFKVeXw%3A1754713332873&q=Biolog%C3%ADa+molecular&sa=X&ved=2ahUKEwjcsuTG8PyOAxWQRjABHSK1ILAQxccNegQIFBAB&mstk=AUtExfBEbxl77INsA-e-TzHk-uFOSBhyP9cjC4mXr4TUN7ClCjuAEsSMj-kb1hWnAoYJQlFu7O5hkkoNXHYMkaM1rKL-lxtVJRmF_iYcYnozCYIr2NyFS-DfillgYir1ChItMAXNMGU1HxgdMafhai9X7RF3E6VsmLkDSYMikQXqb3qAA_X_Up6qkSRIdUbVk8cn8O33hbWE7xDKnzOuVDgnMMYk6UYIRYHn2ReWpDR7GkLFto2jIP_XzktQArncVyV7uSKscc-GpP0D-iXnPHTEUrTw8IKVjbfgUDVt2ybJND47Hg&csui=3)**:** Comparación de secuencias de ADN o proteínas.
* [**Control de versiones**](https://www.google.com/search?client=opera-gx&sca_esv=e8b3de2c0113d5ce&cs=1&sxsrf=AE3TifNr-IS48Bl-HpCQ5YHWDlWtFKVeXw%3A1754713332873&q=Control+de+versiones&sa=X&ved=2ahUKEwjcsuTG8PyOAxWQRjABHSK1ILAQxccNegQIGBAB&mstk=AUtExfBEbxl77INsA-e-TzHk-uFOSBhyP9cjC4mXr4TUN7ClCjuAEsSMj-kb1hWnAoYJQlFu7O5hkkoNXHYMkaM1rKL-lxtVJRmF_iYcYnozCYIr2NyFS-DfillgYir1ChItMAXNMGU1HxgdMafhai9X7RF3E6VsmLkDSYMikQXqb3qAA_X_Up6qkSRIdUbVk8cn8O33hbWE7xDKnzOuVDgnMMYk6UYIRYHn2ReWpDR7GkLFto2jIP_XzktQArncVyV7uSKscc-GpP0D-iXnPHTEUrTw8IKVjbfgUDVt2ybJND47Hg&csui=3)**:** Implementación de comandos como diff para comparar archivos.
* [**Procesamiento de lenguaje natural**](https://www.google.com/search?client=opera-gx&sca_esv=e8b3de2c0113d5ce&cs=1&sxsrf=AE3TifNr-IS48Bl-HpCQ5YHWDlWtFKVeXw%3A1754713332873&q=Procesamiento+de+lenguaje+natural&sa=X&ved=2ahUKEwjcsuTG8PyOAxWQRjABHSK1ILAQxccNegQIFhAB&mstk=AUtExfBEbxl77INsA-e-TzHk-uFOSBhyP9cjC4mXr4TUN7ClCjuAEsSMj-kb1hWnAoYJQlFu7O5hkkoNXHYMkaM1rKL-lxtVJRmF_iYcYnozCYIr2NyFS-DfillgYir1ChItMAXNMGU1HxgdMafhai9X7RF3E6VsmLkDSYMikQXqb3qAA_X_Up6qkSRIdUbVk8cn8O33hbWE7xDKnzOuVDgnMMYk6UYIRYHn2ReWpDR7GkLFto2jIP_XzktQArncVyV7uSKscc-GpP0D-iXnPHTEUrTw8IKVjbfgUDVt2ybJND47Hg&csui=3)**:** Análisis de similitud de textos.
* **Detección de plagio:** Identificación de contenido copiado.

La **Distancia de Edición** (o distancia de Levenshtein) es un problema clásico en informática que consiste en encontrar el número mínimo de operaciones necesarias para transformar una secuencia en otra. Las operaciones permitidas suelen ser: **inserción**, **eliminación** y **sustitución** de caracteres.  
Si tienes las secuencias **"GATO"** y **"PATO"**, la distancia de edición es **1**, ya que basta con sustituir la letra **G** por **P** para convertir una en la otra.

**Aplicaciones:**  
• Procesamiento de lenguaje natural: Corrección ortográfica y búsqueda aproximada de palabras.  
• Reconocimiento de voz: Ajuste de coincidencias entre palabras pronunciadas y palabras del diccionario.

**El Problema de la Mochila** es un problema clásico de optimización combinatoria que consiste en seleccionar un conjunto de elementos con diferentes valores y pesos, de forma que el valor total sea lo mayor posible sin exceder una capacidad máxima establecida para la mochila. Existen diferentes variantes, siendo la más común la Mochila 0/1, donde cada elemento solo puede ser incluido o no, sin fraccionarse.

Si tienes una mochila con capacidad de 50 kg y una lista de objetos con sus respectivos pesos y valores, el objetivo es determinar qué combinación de objetos maximiza el valor total sin sobrepasar el límite de peso.

Aplicaciones:  
• Optimización de recursos: Selección de proyectos o inversiones bajo un presupuesto limitado.  
• Logística: Carga óptima de contenedores o transporte.  
• Planificación: Asignación de tareas o actividades bajo restricciones de tiempo.  
• Videojuegos: Elección de equipamiento con restricciones de peso o espacio.