

Valentina Montenegro Quevedo
Nicolas Jiménez
Nicolás Sánchez

PROBLEMAS DE PROGRAMACIÓN DINÁMICA

Estructuras de datos no lineales



Índice de **C O N T E N I D O S**

- | | |
|---|-----------------------------------|
| 01. Historia | 05. Problema de la mochila |
| 02. Programación dinámica | 06. Conclusión |
| 03. Subsecuencia común más larga | |
| 04. Distancia de edición | |

HISTORIA

En los años 50, Richard Bellman creó la programación dinámica mientras trabajaba en la RAND Corporation, buscando resolver problemas complejos dividiéndolos en partes más simples. Este enfoque se popularizó en los 80 y 90 con algoritmos como Floyd-Warshall y Viterbi, esenciales en rutas óptimas, telecomunicaciones y biología.

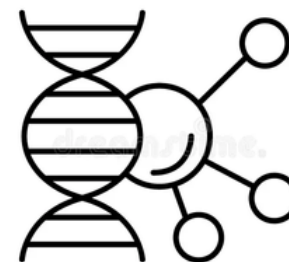
Hoy es clave en IA, bioinformática, economía y más, impulsando avances desde el reconocimiento de voz hasta el análisis genético. El legado de Bellman sigue guiando la ciencia hacia resolver problemas cada vez más grandes y ambiciosos.



Años 50

Creador del método en RAND

Corporation



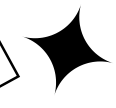
Años 80 - 90

Floyd-Warshall y Viterbi



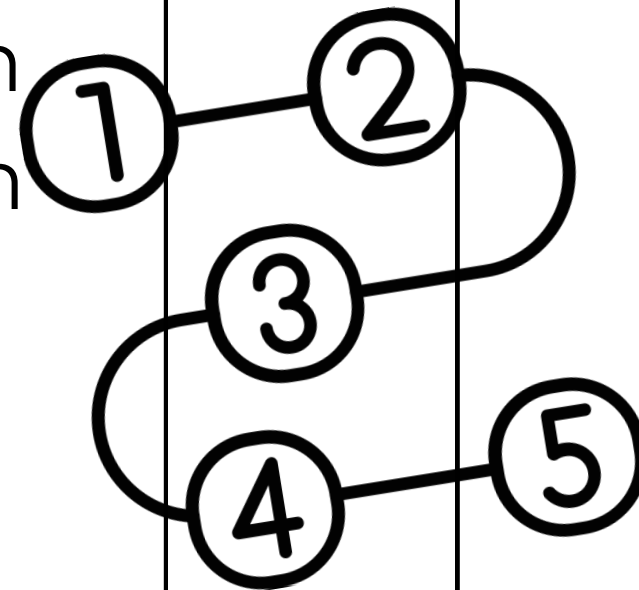
Hoy

Aplicaciones en múltiples disciplinas

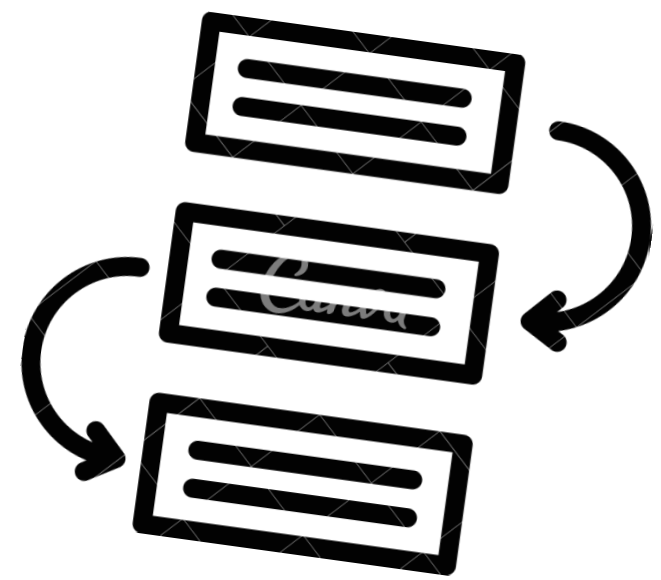


PROGRAMACIÓN DINÁMICA

La programación dinámica tiene como objetivo encontrar una solución de un problema de optimización en forma secuencial. Almacena las soluciones de estos subproblemas para evitar recalcularlas, optimizando así el tiempo de ejecución.



Esta resuelve problemas de optimización al dividirlos en problemas más pequeños, resolviendo cada subproblema y almacenando sus soluciones para poder reutilizarlas y combinarlas para resolver el problema más grande.



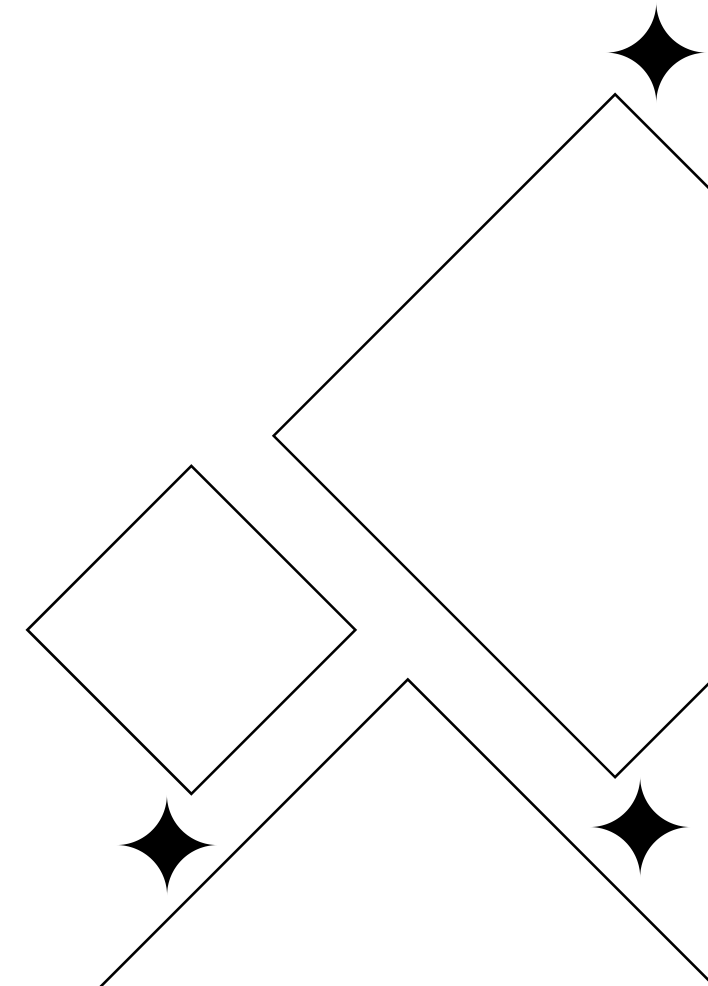
F U N C I O N A M I E N T O

01 Definición de subproblemas: Identificar y dividir etapas, estados y variables.

02 Resolver los subproblemas: Resolver el subproblema identificado, implementando la recursividad o la iteración.

03 Almacenar las soluciones: Las soluciones se almacenan para poder reutilizarlas.

04 Construir la solución al problema original: La solución se construye a partir de los subproblemas que ya se han calculado.





ALGORITMOS DE SOLUCIÓN

LA SUBSECUENCIA COMÚN MAS LARGA

El algoritmo de la Subsecuencia Común Más Larga (LCS, por sus siglas en inglés) encuentra la secuencia de elementos que aparece en el mismo orden en dos cadenas, aunque no necesariamente de forma continua. Es muy usado en comparación de textos, control de versiones y bioinformática. Utiliza programación dinámica para construir una tabla que guarda soluciones parciales, permitiendo obtener la subsecuencia común de forma eficiente.

Ejemplo:

A G G T A B

G X T X A Y B

G T A B

Es la secuencia más larga que aparece en ambas, aunque no de forma consecutiva.

DISTANCIA DE EDICIÓN

El algoritmo de Distancia de Edición (o distancia de Levenshtein) mide el número mínimo de operaciones necesarias para transformar una cadena en otra. Las operaciones permitidas suelen ser inserción, eliminación y sustitución de caracteres. Usando programación dinámica, se construye una tabla donde cada celda representa el costo mínimo para convertir un prefijo de la primera cadena en un prefijo de la segunda, permitiendo resolver el problema de forma eficiente.

Ejemplo:

G A T O

P A T O

G → **P**

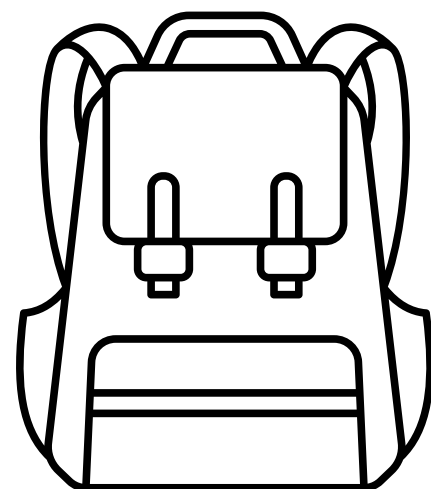
En este caso se sustituye la G por la P.

La distancia de edición es 1, porque con una sola sustitución las cadenas son idénticas.

PROBLEMA DE LA MOCHILA

El Problema de la Mochila (Knapsack Problem) es un clásico de la programación dinámica que consiste en seleccionar un conjunto de objetos con valores y pesos dados para colocarlos en una mochila con capacidad limitada, de forma que el valor total sea máximo sin exceder el peso permitido. Este problema tiene muchas aplicaciones en logística, finanzas y optimización de recursos, y se resuelve construyendo una tabla que combina decisiones óptimas para subproblemas más pequeños.

Ejemplo:



Capacidad de la mochila: 10 kg

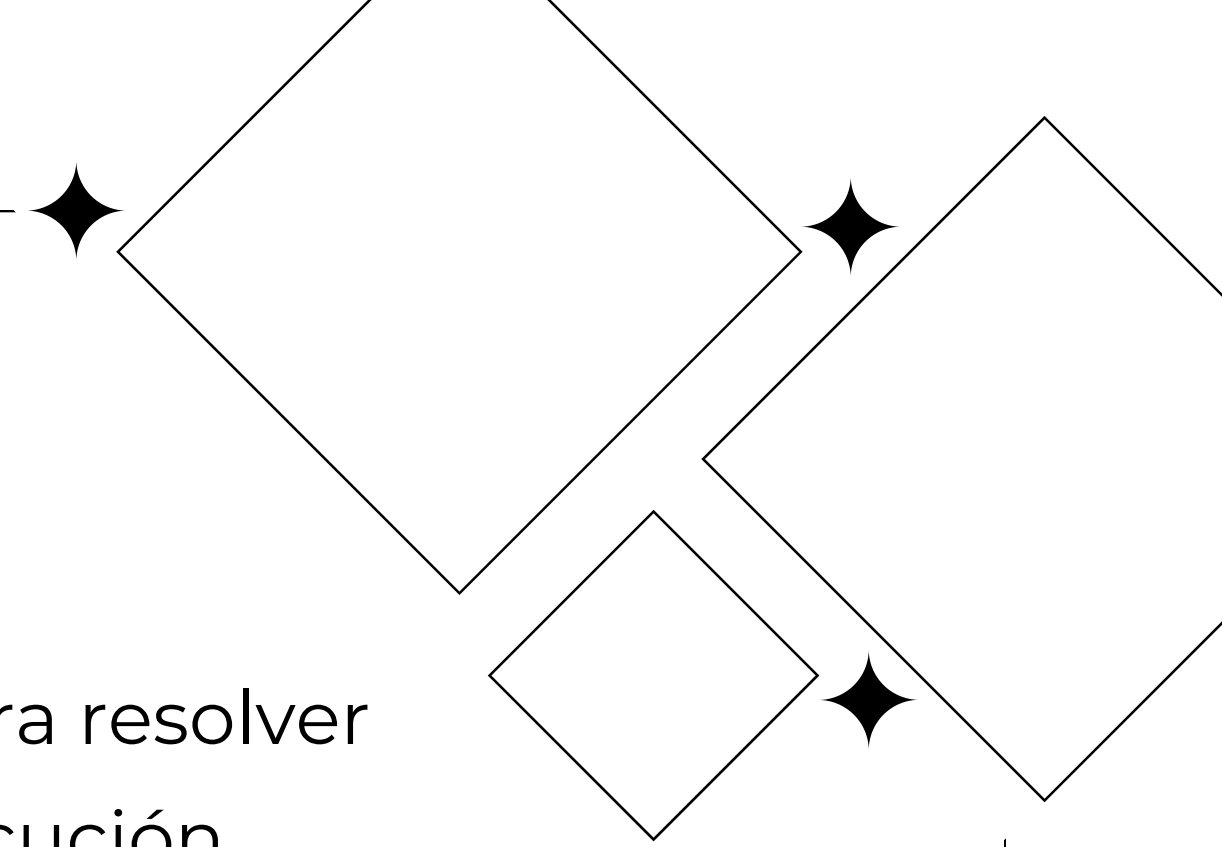


Objeto	Peso kg	Valor
Libro	3	60
Botella	2	100
Chaqueta	5	150
Audifonos	4	120

Mejor combinación = Botella + Audifonos + Libro
Peso 9 kg - Valor total 280

CONCLUSIÓN

La programación dinámica es una herramienta poderosa para resolver problemas complejos optimizando recursos y tiempo de ejecución. Algoritmos como el LCS y el problema de la mochila demuestran cómo, al descomponer un problema en subproblemas más pequeños y reutilizar soluciones previas, se puede encontrar la mejor respuesta de forma eficiente. Estos métodos no solo simplifican el proceso de cálculo, sino que también garantizan resultados óptimos en situaciones reales.



A decorative frame composed of thin black lines. It features a horizontal line at the top and bottom, and a vertical line on the left and right. The corners are adorned with clusters of three diamonds of varying sizes, each with a small four-pointed star at its center. The diamonds are arranged in a way that they appear to be floating or attached to the frame lines.

**GRACIAS POR
SU ATENCIÓN**