

1ER PARCIAL DE INTELIGENCIA ARTIFICIAL (PARTE 9)

NOMBRE: CHOQUE CALLISAYA FERNANDO HUGO

El problema de la mochila es algo común que podemos encontrar en la vida cotidiana pero también en el ámbito de la programación, consiste en intentar maximizar el número de artículos que puedes meter en la columna, en esta ocasión decidimos optar por el algoritmo de la mochila (DCP) para resolver el problema que nos piden en el examen.

El Algoritmo de Programación Dinámica de la Mochila, que es un enfoque eficiente para resolver el problema de la mochila, con el requerimiento de suponer que los elementos y la capacidad de la mochila son relativamente pequeños.

DEFINICION DEL ALGORITMO DCP

La programación dinámica es un método que se utiliza para resolver problemas complejos dividiéndolos en subproblemas más simples. La idea es resolver cada subproblema una vez y almacenar su resultado para que no sea necesario volver a resolverlo.

Necesitaremos estos componentes para resolver este problema:

- Artículos. - Cada artículo tendrá un peso y un valor.
- Peso Total.- El peso total que puede soportar cada mochila.
- Función de estado.- Define cómo se representa la solución en términos de subproblemas.

Respecto a la notación, usaremos n para el número de artículos, W la capacidad máxima de la mochila, $w[i]$ será el peso de algún artículo i , $v[i]$ el valor del artículo i , y por último $K[i][j]$ que será el valor máximo que se pueda obtener por los primeros i artículos y una capacidad de mochila j .

¿COMO FUNCIONA EL ALGORITMO?

- Crearemos la tabla $K[i][j]$
- Inicializaremos $K[0][j] = 0$ para todo j (si no hay artículos, el valor es 0) y $K[i][0] = 0$ para todo i (si la capacidad de la mochila es 0, el valor también es 0).
- Recorreremos cada artículo i desde 1 hasta n y para cada capacidad j desde 1 hasta W .

Aquí podemos recorrer dos caminos:

- Si el peso del artículo i es menor o igual a j , calculamos el valor máximo, incluiremos el artículo si el valor es $v[i - 1] + K[i - 1][j - w[i - 1]]$.

- Si el peso del artículo i es mayor que j , se copia el valor sin incluir el artículo $K[i][j] = K[i - 1][j]$.
- El valor máximo que se puede obtener con n artículos y capacidad W se encuentra en $K[n][W]$.
- Para determinar que artículos se incluyen en la solución óptima, se puede hacer un recorrido inmerso en la tabla.
- Comienza desde $K[n][W]$ y verifica si el valor es diferente de $K[n - 1][W]$, si es diferente, significa que el artículo n está incluido, restamos el peso de W y decrementamos n .

Si es igual, significa que el artículo no está incluido, por lo cual solo decrementamos n .

EJEMPLO (CON CHATGPT)

Supongamos que tenemos 3 artículos con los siguientes pesos y valores:

Artículo	Peso	Valor
1	1	1
2	4	5
3	3	4

La capacidad (W) de la mochila es 4.

1. Inicializamos la tabla K de tamaño 4×5
2. Llenamos la tabla usando las reglas de arriba.
3. Al final $K[3][4]$ nos dará el valor máximo que podemos obtener.

Esta es una técnica poderosa para resolver el problema de la mochila, ya que permite dividir el problema en subproblemas más manejables y almacenar los resultados, lo que reduce la complejidad en comparación con otros enfoques más simples. Esta técnica es especialmente útil en aplicaciones donde se requiere una solución óptima y se pueden manejar las restricciones de espacio y tiempo.