INSTITUTO POLITÉCNICO NACIONAL

Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas.

MATERIA:

- Analisis y diseño de algoritmos

MAESTRA:

- Erika Sánchez Femat

ALUMNA:

- Naylin Yusseth Martínez Mireles

PROGRAMACIÓN DINAMICA

Índice

1. INTRODUCCIÓN

1.1. ¿QUÉ ES?

La programación dinámica es una técnica de resolución de problemas en informática y matemáticas que se utiliza para optimizar la eficiencia de algoritmos mediante la descomposición de un problema en subproblemas más pequeños y resolviéndolos de manera recursiva.
A diferencia de enfoques más ingenuos que resuelven los mismos subproblemas una y otra vez, la programación dinámica almacena las soluciones de subproblemas ya resueltos y reutiliza esas soluciones para evitar redundancias y mejorar la eficiencia.

La programación dinámica es especialmente útil en problemas que exhiben la propiedad de solapamiento de subproblemas y la propiedad de optimalidad, lo que significa que la solución óptima global puede construirse a partir de soluciones óptimas a subproblemas más pequeños.

Esta técnica se aplica comúnmente en una variedad de campos, como la optimización de algoritmos, la teoría de juegos, la economía, la bioinformática y muchos otros. Uno de los problemas clásicos resueltos con programación dinámica es el problema de la mochila, pero hay muchos otros ejemplos que abarcan una amplia gama de dominios.
En resumen, la programación dinámica es una herramienta poderosa para abordar problemas complejos y mejorar la eficiencia computacional mediante la reutilización de soluciones a subproblemas.

2. DESARROLLO

2.1. ¿CÓMO FUNCIONA?

La programación dinámica funciona dividiendo un problema en subproblemas más pequeños y resolviéndolos de manera recursiva. La clave es que, una vez que se resuelve un subproblema, su solución se almacena y se reutiliza en caso de que el mismo subproblema vuelva a surgir.
Esto evita la redundancia y mejora la eficiencia del algoritmo.

Hay dos propiedades fundamentales que deben cumplirse para que la programación dinámica sea aplicable:

- Solapamiento de Subproblemas (Overlapping Subproblems): El problema original puede dividirse en subproblemas más pequeños que se resuelven de forma independiente. Además, estos subproblemas comparten soluciones, lo que significa que la misma subinstancia del problema se resuelve varias veces.

- Subestructura Óptima (Optimal Substructure): Una solución global óptima se puede construir a partir de soluciones óptimas de subproblemas más pequeños. Esto implica que la solución al problema general se puede obtener combinando las soluciones de los subproblemas de manera eficiente.

El proceso típico de resolución de problemas mediante programación dinámica sigue estos pasos:

1. Divide el Problema:

    - Divide el problema en subproblemas más pequeños y más manejables.

2. Resuelve Subproblemas:

   - Resuelve cada subproblema de forma independiente, ya sea de manera recursiva o iterativa.

3. Almacena Soluciones:

   - Almacena las soluciones de los subproblemas resueltos en una tabla o memoria para evitar recálculos innecesarios.

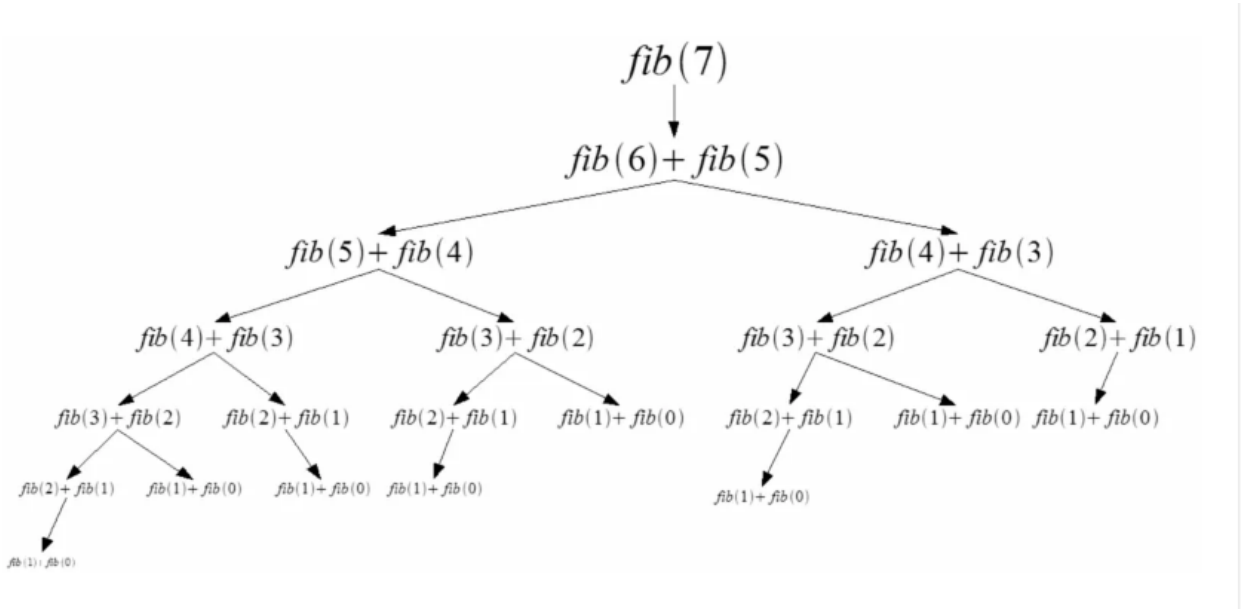
4. Combina Soluciones:

   - Combina las soluciones de los subproblemas para obtener la solución al problema original.

La memorización (o memoización) es una técnica comúnmente utilizada en programación dinámica para almacenar y recuperar las soluciones de subproblemas de manera eficiente.
Además, la programación dinámica puede implementarse de manera ascendente (bottom-up) o descendente (top-down), dependiendo de si se comienza resolviendo los subproblemas más pequeños o desde el problema original.

Ejemplo 1.
Un ejemplo clásico de aplicación de programación dinámica es el algoritmo de la secuencia de Fibonacci, donde la recursión ingenua resulta en muchos cálculos redundantes, pero la programación dinámica mejora significativamente la eficiencia al almacenar y reutilizar las soluciones de los subproblemas.

El codigo fibonacci ya lo hemos analizado, asi que no entraremos tanto en como funciona, solo resaltar que este seria un ejemplo de practica dinamica.



2.2. IDENTIFICACION DE LA PRACTICA DINAMICA

Identificar si un problema se puede resolver mediante programación dinámica puede implicar el reconocimiento de ciertas características y propiedades del problema. Aquí hay algunas señales que podrían indicar que un problema es susceptible de ser resuelto mediante programación dinámica:

. Solapamiento de Subproblemas (Overlapping Subproblems):
Si el problema puede dividirse en subproblemas más pequeños y la solución del problema original involucra la solución repetida de esos subproblemas, existe un solapamiento de subproblemas. La reutilización de soluciones a subproblemas es fundamental en la programación dinámica.

. Subestructura Óptima (Optimal Substructure):
La propiedad de subestructura óptima significa que una solución global óptima puede construirse a partir de soluciones óptimas a subproblemas más pequeños. Es decir, la solución al problema general se puede obtener combinando las soluciones de subproblemas de manera eficiente.

. Memorización o Tabulación:
· **Memorización (Top-Down):** Si al resolver el problema recursivamente, se observa que se están resolviendo los mismos subproblemas varias veces, la memorización (o memoización) puede ayudar a almacenar y reutilizar las soluciones anteriores para evitar recalculos innecesarios.
· **Tabulación (Bottom-Up):** Si se puede construir una tabla y llenarla iterativamente para almacenar soluciones a subproblemas más pequeños, es un indicativo de que la programación dinámica podría ser aplicable.

. Elección Óptima de Subestructuras Locales:
La resolución de un problema mediante programación dinámica a menudo implica tomar decisiones óptimas en subestructuras locales para llegar a una solución global óptima.

. Estructura Recurrente:

· Si al abordar el problema, hay una estructura recurrente que se puede expresar matemáticamente, es posible que la programación dinámica sea apropiada.

· Por ejemplo, relaciones de recurrencia que expresan la solución en función de soluciones más pequeñas del mismo problema.

. Optimalidad de la Solución:

· Si el problema satisface la propiedad de optimalidad, donde la solución óptima del problema global se puede construir a partir de soluciones óptimas de subproblemas, es probable que la programación dinámica sea aplicable.

2.3. CONCEPTOS QUE UTILIZA LA PROGRAMACION DINAMICA

Algunas conclusiones clave sobre la programación dinámica son las siguientes:

1. Divide y Vencerás: La programación dinámica se basa en la idea de dividir un problema grande en subproblemas más pequeños y resolverlos de manera independiente. Esto facilita la comprensión y la resolución del problema global.
2. Optimalidad y Solapamiento: La programación dinámica se aplica especialmente cuando un problema exhibe la propiedad de optimalidad y solapamiento de subproblemas. La solución óptima global puede construirse a partir de soluciones óptimas a subproblemas más pequeños, y estos subproblemas comparten soluciones.
3. Memorización y Tabulación: La técnica puede implementarse de dos maneras: mediante memorización (arriba hacia abajo) y tabulación (abajo hacia arriba). Ambos enfoques buscan evitar recalculos innecesarios almacenando y reutilizando soluciones a subproblemas.

4. Problemas Variados: La programación dinámica se aplica en una amplia gama de problemas, desde optimización de algoritmos hasta problemas en economía, bioinformática, inteligencia artificial y más. Algunos ejemplos comunes incluyen el problema de la mochila, el problema de la subsecuencia común más larga y el problema del camino más corto en grafos ponderados.

5. Complejidad Temporal y Espacial: Aunque la programación dinámica puede mejorar la eficiencia al evitar cálculos redundantes, es esencial considerar la complejidad temporal y espacial asociada con la construcción de tablas y la memoria utilizada.

6. Habilidad en el Diseño de Algoritmos: Identificar problemas que se prestan a la programación dinámica y diseñar algoritmos eficientes requiere habilidad y práctica. Analizar la estructura del problema y reconocer propiedades clave son habilidades fundamentales en la aplicación exitosa de esta técnica.

3. CONCLUSIÓN

En conclusión, la programación dinámica es una técnica poderosa y versátil utilizada en ciencias de la computación y matemáticas para resolver problemas complejos y optimizar algoritmos. Su enfoque se basa en dividir un problema en subproblemas más pequeños, resolverlos de manera eficiente y almacenar las soluciones para evitar recálculos innecesarios.

En resumen, la programación dinámica es una herramienta valiosa en el arsenal de un programador o científico de datos cuando se enfrenta a problemas que pueden descomponerse en subproblemas más pequeños y compartan soluciones. La capacidad de aplicar eficazmente la programación dinámica puede conducir a soluciones eficientes y elegantes para una variedad de desafíos computacionales.

4. REFERENCIAS

https://geekflare.com/es/dynamic-programming/
https://www.lifeder.com/programacion-dinamica/
https://historiadelaempresa.com/programacion-dinamica
https://www.ingenieria.unam.mx/sistemas/PDF/Avisos/Seminarios/SeminarioV/Sesion6_IdaliaFlores_20abr15.pdf
https://minerva.usc.es/xmlui/bitstream/handle/10347/28920/L%C3%B3pez_Fern%C3%A1ndez_Lidia.pdf?sequence=1