Desarrollar un programa que permita analizar y maximizar la ganancia de compra y venta de acciones utilizando programación dinámica. Contexto: Un inversionista quiere comprar y vender acciones de una empresa con un número máximo de transacciones K. Dado un conjunto de precios históricos de la acción, debe determinar la máxima ganancia posible siguiendo estas reglas: Puede comprar y vender la acción hasta K veces. No puede comprar sin haber vendido antes una acción anterior. Debe encontrar la mejor estrategia para maximizar su ganancia.

Paso 1: Definir el problema

Tenemos un array precios que representa los precios históricos de las acciones en diferentes días. Queremos encontrar la máxima ganancia posible realizando como máximo K transacciones (compra y venta).

Paso 2: Diseñar la solución

Utilizaremos una tabla dinámica dp donde dp[i][j] representará la máxima ganancia hasta el día i con j transacciones. La relación de recurrencia será:

Si no realizamos ninguna transacción en el día i, entonces dp[i][j] = dp[i-1][j].

Si realizamos una transacción en el día i, entonces dp[i][j] = max(dp[i][j], dp[m][j-1] + precios[i] - precios[m]) para todo m < i.

**Implementacion Java**

public class MaximizadorGanancias {

public static int maximizarGanancias(int[] precios, int k) {

int n = precios.length;

if (n == 0 || k == 0) {

return 0;

}

// Si k es mayor o igual a n/2, podemos realizar todas las transacciones posibles

if (k >= n / 2) {

int gananciaMaxima = 0;

for (int i = 1; i < n; i++) {

if (precios[i] > precios[i - 1]) {

gananciaMaxima += precios[i] - precios[i - 1];

}

}

return gananciaMaxima;

}

// Tabla dinámica para almacenar las ganancias máximas

int[][] dp = new int[k + 1][n];

// Llenar la tabla dinámica

for (int transaccion = 1; transaccion <= k; transaccion++) {

int maxGananciaAnterior = -precios[0];

for (int dia = 1; dia < n; dia++) {

dp[transaccion][dia] = Math.max(dp[transaccion][dia - 1], precios[dia] + maxGananciaAnterior);

maxGananciaAnterior = Math.max(maxGananciaAnterior, dp[transaccion - 1][dia] - precios[dia]);

}

}

// La máxima ganancia estará en dp[k][n-1]

return dp[k][n - 1];

}

public static void main(String[] args) {

int[] precios = {3, 2, 6, 5, 0, 3};

int k = 2;

int gananciaMaxima = maximizarGanancias(precios, k);

System.out.println("La máxima ganancia con " + k + " transacciones es: " + gananciaMaxima);

}

}