Actividad 2 Objetivo: Dada una lista de números, encontrar los dos números mayores utilizando la técnica de Divide y Conquista. Tareas: Pseudocódigo: Escribe el pseudocódigo que resuelva el problema. Implementación en Java: Implementa el pseudocódigo en Java

Función encontrarDosMayores(lista, inicio, fin):

Si inicio == fin:

Devolver (lista[inicio], -∞) // Caso base: un solo elemento

Si fin - inicio == 1:

Si lista[inicio] > lista[fin]:

Devolver (lista[inicio], lista[fin])

Sino:

Devolver (lista[fin], lista[inicio])

medio = (inicio + fin) / 2 // Dividir la lista en dos mitades

// Resolver recursivamente para cada mitad

(maxIzquierda, segundoMaxIzquierda) = encontrarDosMayores(lista, inicio, medio)

(maxDerecha, segundoMaxDerecha) = encontrarDosMayores(lista, medio + 1, fin)

// Combinar: encontrar los dos mayores entre las dos mitades

Si maxIzquierda > maxDerecha:

max1 = maxIzquierda

max2 = max(segundoMaxIzquierda, maxDerecha)

Sino:

max1 = maxDerecha

max2 = max(segundoMaxDerecha, maxIzquierda)

Devolver (max1, max2)

**Implementacion Java**

class DosMayoresDivideYConquista {

public static void main(String[] args) {

int[] lista = {10, 5, 20, 15, 30, 25}; // Lista de números

int[] resultado = encontrarDosMayores(lista, 0, lista.length - 1);

System.out.println("Los dos números mayores son: " + resultado[0] + " y " + resultado[1]);

}

// Método para encontrar los dos números mayores usando Divide y Conquista

public static int[] encontrarDosMayores(int[] lista, int inicio, int fin) {

// Caso base: si solo hay un elemento

if (inicio == fin) {

return new int[]{lista[inicio], Integer.MIN\_VALUE};

}

// Caso base: si hay dos elementos

if (fin - inicio == 1) {

if (lista[inicio] > lista[fin]) {

return new int[]{lista[inicio], lista[fin]};

} else {

return new int[]{lista[fin], lista[inicio]};

}

}

// Dividir la lista en dos mitades

int medio = inicio + (fin - inicio) / 2;

// Resolver recursivamente para cada mitad

int[] maxIzquierda = encontrarDosMayores(lista, inicio, medio);

int[] maxDerecha = encontrarDosMayores(lista, medio + 1, fin);

// Combinar: encontrar los dos mayores entre las dos mitades

int max1, max2;

if (maxIzquierda[0] > maxDerecha[0]) {

max1 = maxIzquierda[0];

max2 = Math.max(maxIzquierda[1], maxDerecha[0]);

} else {

max1 = maxDerecha[0];

max2 = Math.max(maxDerecha[1], maxIzquierda[0]);

}

return new int[]{max1, max2};

}

}