

URL VIDEOS

Explicación enunciado del problema: https://youtu.be/oOh_PyyXV4?feature=shared

Explicación código: https://www.youtube.com/watch?v=jpta_3PbwHA

Análisis de complejidad: <https://youtu.be/NLu4c0vHIrY>

ENUNCIADO DEL PROBLEMA

Dada una matriz de tamaño N. La tarea consiste en encontrar el elemento mínimo y máximo de la matriz utilizando el mínimo número de comparaciones.

Entradas

Un arreglo de enteros.

Salidas

El valor mínimo y máximo.

URL DE GITLAB

[Ejercicios Proyecto ADA / Ejercicio 12 - Minmax Array: GitLab](#)

MÉTODO ITERATIVO

```
public int[] metodoIterativo(int[] array) {  
    int mayor = array[0], menor = array[0];  
    int[] result = new int[2];  
    for (int i = 1; i < array.length; i++) {  
        if (array[i] < menor) {  
            menor = array[i];  
        }  
        if (array[i] > mayor) {  
            mayor = array[i];  
        }  
    }  
    result[0] = menor;  
    result[1] = mayor;  
    return result;  
}
```

| | |
|-----------------|--|
| Eficacia | El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas. |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO ITERATIVO: 1 9 |

| | |
|------------|--|
| | INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO MÉTODO ITERATIVO: -500000 499995 |
| | INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO ITERATIVO: -9999 9999 |
| Eficiencia | El método es eficiente ya que recorre una sola vez el arreglo para hallar los valores mínimo y máximo, sin embargo en arreglos muy grandes es un poco lento. |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO ITERATIVO: 1 9 Tiempo: 186 microsegundos |
| | INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO MÉTODO ITERATIVO: -500000 499995 Tiempo: 2467 microsegundos |
| | INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO ITERATIVO: -9999 9999 Tiempo: 14 microsegundos |

| | |
|--------------------|---|
| Correctitud | El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO ITERATIVO: 1 9 RESPUESTA ESPERADA: 1 9 |
| | INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO MÉTODO ITERATIVO: -500000 499995 RESPUESTA ESPERADA: -500000 499995 |
| Completitud | INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO ITERATIVO: -9999 9999 RESPUESTA ESPERADA: -9999 9999 |
| | El método es completo, ya que la lógica usada para buscar el mínimo y el máximo es correcta, asegura siempre un resultado correcto, si es que este existe |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO ITERATIVO: 1 9 RESPUESTA ESPERADA: 1 9 |
| | INSTANCIA 2 (ARREGLO GRANDE): |

| | |
|--|---|
| | TAMAÑO ARREGLO: 100000 RESULTADO MÉTODO ITERATIVO: -500000 499995 RESPUESTA ESPERADA: -500000 499995 |
| | INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO ITERATIVO: -9999 9999 RESPUESTA ESPERADA: -9999 9999 |

MÉTODO 2

```
public int[] metodoRecursivo(int[] array) {
    if (array == null || array.length == 0) {
        throw new RuntimeException("El arreglo no puede estar vacío");
    }
    return metodoRecursivo(array, 0, array.length-1);
}

private int[] metodoRecursivo(int[] arr, int primero, int ultimo) {
    if (primero == ultimo) {
        return new int[]{arr[primero], arr[primero]};
    } else if (ultimo == primero + 1) {
        int a = arr[primero];
        int b = arr[ultimo];
        int min, max;
        if (a < b) {
            min = a;
            max = b;
        } else {
            min = b;
            max = a;
        }
        return new int[]{min, max};
    }
    int mid = (primero + ultimo) / 2;
    int[] izquierda = metodoRecursivo(arr, primero, mid);
    int[] derecha = metodoRecursivo(arr, mid + 1, ultimo);

    return compararMinMax(izquierda, derecha);
}

private static int[] compararMinMax(int[] a, int[] b) {
    int min, max;
    if(a[0]<b[0]){
        min = a[0];
    }else{
        min = b[0];
    }
    if(a[1]>b[1]){
        max = a[1];
    }else{
        max = b[1];
    }
    return new int[]{min,max};
}
```

| | |
|-------------------|---|
| Eficacia | El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO RECURSIVO: 1 9 |
| | INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO MÉTODO RECURSIVO: -500000 499995 |
| | INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO RECURSIVO: -9999 9999 |
| Eficiencia | El método es eficiente en entradas no tan grandes, ya que podemos ir dividiendo el arreglo a la mitad y buscando el mínimo y máximo en cada uno, y retornamos los de ambas mitades para realizar una comparación. |
| | INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO MÉTODO RECURSIVO: 1 9 Tiempo: 16 microsegundos |
| | INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 |
| | |

| | |
|--------------------|--|
| | <p>RESULTADO MÉTODO RECURSIVO: -500000 499995 Tiempo: 4547 microsegundos</p> |
| | <p>INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO MÉTODO RECURSIVO: -9999 9999 Tiempo: 9 microsegundos</p> |
| | <p>El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles</p> |
| | <p>INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO ESPERADO: 1 9 RESULTADO MÉTODO RECURSIVO: 1 9</p> |
| Correctitud | <p>INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO ESPERADO: -500000 499995 RESULTADO MÉTODO RECURSIVO: -500000 499995</p> |
| | <p>INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO ESPERADO: -9999 9999 RESULTADO MÉTODO RECURSIVO: -9999 9999</p> |

| | |
|-------------------|---|
| Compleitud | <p>El método es completo, ya que la lógica usada para buscar el mínimo y máximo del arreglo es correcta, y asegura siempre un resultado correcto, si es que este existe</p> |
| | <p>INSTANCIA 1: 3 1 5 2 9 TAMAÑO ARREGLO: 5 RESULTADO ESPERADO: 1 9 RESULTADO MÉTODO RECURSIVO: 1 9</p> |
| | <p>INSTANCIA 2 (ARREGLO GRANDE): TAMAÑO ARREGLO: 100000 RESULTADO ESPERADO: -500000 499995 RESULTADO MÉTODO RECURSIVO: -500000 499995</p> |
| | <p>INSTANCIA 3: -100 0 20 35 1000 9999 -9999 500 TAMAÑO ARREGLO: 8 RESULTADO ESPERADO: -9999 9999 RESULTADO MÉTODO RECURSIVO: -9999 9999</p> |